

INSTITUTO TECNOLÓGICO DE COSTA RICA

ESCUELA DE INGENIERÍA ELECTRÓNICA

SMARTECH S.A.

Prototipo de terminal para control de producción

**Informe de Proyecto de Graduación para optar por el grado de Bachiller
en Ingeniería Electrónica**

Byron Medrano García

Cartago, Junio 2001

DEDICATORIA

A mi madre, por el gran sacrificio que ha hecho para que sus hijos salgan adelante.

AGRADECIMIENTOS

Agradezco el apoyo y la colaboración brindada para la realización de este Proyecto de Graduación al Ing. Juan Carlos Castañeda y al Ing. Edgar Brenes.

RESUMEN

En la actualidad, en una fábrica de maquila es importante tener un control estricto sobre la producción, esto con el fin de mejorar la calidad del producto y la eficiencia del operario.

En una fábrica de este tipo se dividen en varias secciones o módulos. Cada una de estas secciones realiza una parte del proceso total de producción, por ejemplo: la sección de insertar botones, la sección de pegar mangas, la que hace los ojales a las camisas, etc. En cada sección, en el mejor de los casos, hay un supervisor que lleva un registro de la producción diaria de la sección. Este registro consiste en una hoja de papel que contiene una tabla para ser completada por el supervisor con los datos requeridos. Algunos de estos datos son el nombre del operario, código del producto, cantidad de productos defectuosos, y la cantidad de productos no defectuosos, entre otros.

La idea de realizar esta terminal de control de producción, es para obtener fácilmente, información actualizada constantemente de cada una de las secciones de la línea de producción, en el mismo momento que se va generando.

Lo que se pretende es colocar terminales en cada una de las secciones de la fábrica que están conectadas con el computador. El encargado de cada sección interacciona con el módulo, introduciendo los datos requeridos sobre la producción mediante la tarjeta de código de barras y también por el teclado y la pantalla. La terminal almacena la información que posteriormente va a ser solicitada por el computador a través del puerto de comunicaciones.

Palabras claves: Producción, Terminal, Comunicación serie, Código de barras, sensor.

ABSTRACT

Actually, in a assembly's plant is important to have a strict control of the production, this is for improve the quality of the products and the operator's efficiency.

This kind of factory is divided in many sections or modules. Each section does a part of the production, as instead: the insertion of buttons, sew sleeves, do the buttonholes, etc. In each section, there are a supervisor who does a registry of the production of the day. This registry consists of a sheet that contains a table, the operator has to fill in this table with the required data. Some of this information is the operator's name, quantity of production with problems, product's code, quantity of good among others.

The main goal with this terminal is to get easily the information related to the production from each production's line in real time.

We pretend to put terminals in each plant's section, this terminals will be connected to a personal computer. The person in charge of each section uses the terminal for introducing the required data, for this he uses a barcode card or by a keyboard located in each terminal. The terminal keeps the data, then the computer asks for the information using a communication's port.

Key Words: Production, Terminal, Serial Communications, Barcode, Sensor.

INDICE GENERAL

CAPÍTULO 1 INTRODUCCIÓN	1
1.1 Descripción de la empresa.....	1
1.2 Definición del problema y su importancia	3
1.3 Objetivos.....	4
1.3.1. Objetivo general	4
Objetivos específicos.....	4
CAPÍTULO 2 ANTECEDENTES.....	6
2.1 Estudio del problema a resolver	6
2.2 Requerimientos de la empresa	9
2.3 Solución propuesta	11
CAPÍTULO 3 PROCEDIMIENTO METODOLÓGICO	13
CAPÍTULO 4 DESCRIPCIÓN DEL HARDWARE UTILIZADO	16
CAPÍTULO 5 DESCRIPCIÓN DEL SOFTWARE UTILIZADO	22
CAPÍTULO 6 ANÁLISIS Y RESULTADOS.....	23
6.1 Explicación del diseño.....	23
6.2 Alcances y limitaciones.....	36
CAPÍTULO 7 CONCLUSIONES Y RECOMENDACIONES	37
CONCLUSIONES	37
RECOMENDACIONES	38
BIBLIOGRAFÍA	39
APÉNDICES	40

Apéndice 1: Resumen de las instrucciones utilizadas en el programa	40
Apéndice 2: Funciones de las teclas que presenta el WinIDE Development kit	42
Apéndice 3: Gráficos de diferentes sensores con diferentes leds	43
Apéndice 5: Glosario.....	45
ANEXOS	46
Anexo 1: Características del transceiver LTC1487	46

INDICE DE FIGURAS

Figura 1.1. Diagrama de bloques del proceso a implementar en una fábrica	8
Figura 2.1 Diagrama de bloques general de la terminal.	11
Figura 4.1 Diagrama de bloques del teclado utilizado	16
Figura 4.2 Diagrama matricial del teclado utilizado	17
Figura 4.3 Bloque funcional de la pantalla.....	18
Figura 6.1. Formato ITF para las tarjetas	24
Figura 6.2. Diagrama de flujo de lectura del teclado	25
Figura 6.3. Continuación diagrama de flujo de lectura del teclado	26
Figura 6.4. Continuación diagrama de flujo de lectura del teclado	27
Figura 6.5. Continuación diagrama de flujo de lectura del teclado	28
Figura 6.6. Diagrama de flujo de escritura de las teclas	30
Figura 6.7. Continuación diagrama de flujo de lectura del teclado	31

INDICE DE TABLAS

Tabla 4.1 Definición de los pines de salida del teclado.....	17
Tabla 4.2 Puertos del microcontrolador MC68HC908GP32	20
TABLA A1 Resumen de las instrucciones utilizadas en el programa	40

CAPÍTULO 1

INTRODUCCIÓN

Lo que se pretende con este proyecto es implementar un sistema que realice automáticamente el proceso de recolección de la información de cada una de las secciones de una fábrica de maquila, y lo almacene en un computador para su posterior procesamiento.

Así, se pretende con este sistema, corregir la deficiencia de información que llega tarde y de manera muy tediosa a las manos de las personas encargadas de velar por la producción y el buen rendimiento de la compañía; de tal forma que la empresa corrija inmediatamente las fallas y prevengan futuros problemas.

Debido a que es un proyecto que se va a producir en masa, la empresa a puesto ciertas restricciones con la información, ya que son de índole confidencial para la misma y por ende en este informe no se presentan esquemas de circuitos, software y tampoco la relevación de algunos componentes que son claves para esta terminal.

1.1 Descripción de la empresa

SMARTECH es una empresa nueva que recién fue fundada en el año 2000; su fundador fue el presidente de la misma el Ing. Juan Carlos Castañeda.

La empresa se encuentra ubicada en el centro comercial La Carpintera, en Residencial Los Pinares, localizado en carretera vieja a Tres Ríos.

La empresa nació con la idea de incursionar en el mercado con diseños electrónicos propios, para brindar facilidades a diferentes empresas en sus procesos de producción y además de automatizar con componentes electrónicos mecanismos que requieren ser más flexibles y de mejor accesibilidad al operador.

El objetivo principal de la empresa es brindar soluciones por medio de diseños propios y del menor costo posible a los problemas de producción, que viven hoy en

día diferentes empresas y fábricas, a nivel nacional, Centroamérica, México, el Caribe y Sudamérica.

Actualmente estos diseños se realizan por medio de microcontroladores de Motorola debido a su gran flexibilidad de manejo y operación, que ligados a diferentes dispositivos electrónicos producen una buena solución para el cliente.

Con el uso de los microcontroladores de Motorola se hace posible la minimización de componentes para los diferentes sistemas a implementar. Además existe un representante local de productos de Motorola, esto facilita de alguna manera la adquisición rápida de información y de sus productos.

Al ser una empresa que apenas está dando sus primeros pasos no cuenta con muchos empleados, en la actualidad laboran en la misma 3 personas de los cuales son dos ingenieros en electrónica y un administrador.

El proyecto se desarrolla en el área de diseño electrónico, del cual está a cargo el mismo presidente, el Ing. Juan Carlos Castañeda. Don Juan Carlos es ingeniero en electrónica, con grado de bachiller, graduado del ITCR hace 15 años. Cuenta con vasta experiencia en el campo de los microcontroladores y equipo electrónico.

En esta área se realizan varios procesos para llegar a la solución final. Primero que todo se hace un estudio del mercado al cual se va a colocar el producto, luego habiendo estudiado las necesidades del cliente se diseña el circuito a implementar, después de revisarlo y hacerle correcciones, para disminuir costos y mejorar la eficiencia, se ensambla y se hacen las pruebas respectivas al prototipo.

El microcontrolador debe ser programado según las necesidades del problema a solucionar.

En la parte de diseño se hace también el circuito impreso, mediante un software especializado para tal objetivo, que posteriormente se manda a fabricar para que quede listo para el ensamble de los componentes.

Luego de haber hecho el respectivo estudio se elabora la lista de materiales a utilizar y se procede a la respectiva compra.

Una vez completados todos los procedimientos de desarrollo, se proceden a ensamblar los componentes y después a realizar las pruebas finales del producto en desarrollo.

Posteriormente, cuando está listo el producto, pasa a la etapa de validación por parte de la empresa que consiste en aprobar el prototipo, para luego mandarlo al mercado.

1.2 Definición del problema y su importancia

La forma de elaborar un producto en la empresa de maquila de ropa por ejemplo es por medio de líneas de producción donde se encuentra el personal que realiza procesos tales como costura, ojales, inserción de botones y finalmente la etapa de control de calidad.

El proceso de control de calidad se realiza cuando se ha terminado de ensamblar el producto y se encuentra en la etapa de revisión, ahí se separa el producto bueno del defectuoso, cada cierto periodo el supervisor de la línea debe llenar una boleta donde se anota el mínimo de piezas que se consideran buenas y cuantas presentaron errores durante la revisión, además supervisa las labores del operario, que redundan en la eficiencia de su producción

Al final del día cada supervisor de línea lleva esas boletas al encargado de producción para que ahí se lleve el registro diario.

Este registro consiste en el mejor de los casos, en una de base de datos, en la cual se tienen que ingresar los datos manualmente, ahí se lleva un historial de los operarios con respecto a su productividad y eficiencia.

El problema se presenta en caso de haber algún problema durante la producción por alguna falla en el proceso o un mal procedimiento en alguna de las etapas, este

error se detecta hasta la mañana siguiente cuando se revisan las boletas del día anterior, causando pérdidas por el material utilizado en esas piezas y pérdidas por el tiempo invertido por los operarios.

Además, siempre y cuando sea posible, el departamento de producción intenta reparar estas piezas defectuosas utilizando nuevamente todo el material que se pueda aprovechar en otras piezas o trata de reparar el daño cuando es pequeño y la reparación no afecta la calidad del producto.

Para reparar estas piezas se invierte tiempo del operario encargado de esta labor, en algunos casos el dispositivo requiere de revisión exhaustiva por parte de un supervisor capacitado, lo que lleva a que el mismo invierta parte de su tiempo en estas reparaciones, utilizando equipo y materiales que se podrían utilizar en otras tareas.

1.3 Objetivos

1.3.1. Objetivo general

Implementar, programar y probar, para la empresa SMARTECH, un prototipo de terminal de control de producción para plantas de maquila.

Objetivos específicos

1.3.2. Objetivos específicos relacionados con el MCU:

1.3.2.1. Conocer el funcionamiento del microcontrolador de Motorola 68HC908GP32.

1.3.2.2. Investigar el funcionamiento de las tarjetas con códigos de barra.

1.3.2.3. Conocer el programa de programación y emulación de Motorola WinIDE.

1.3.2.4. Conocer el hardware del emulador la familia HC08 de Motorola.

1.3.2.5. Hacer pruebas con el emulador de Motorola.

1.3.3. Objetivos específicos relacionados con el diseño de la terminal:

1.3.3.1. Estudiar el diseño actual del circuito del prototipo a implementar.

1.3.3.2. Conocer el funcionamiento de los sensores ópticos tipo CCD lineal.

1.3.3.3. Conocer el circuito integrado de comunicación RS-485.

1.3.3.4. Conocer la pantalla de cristal liquido.

1.3.3.5. Implementar el hardware del prototipo terminal de control de producción.

1.3.4. Objetivos específicos relacionados con el software:

1.3.4.1 Implementar el software del microcontrolador de Motorola 68HC908GP32.

CAPÍTULO 2

ANTECEDENTES

2.1 Estudio del problema a resolver

Primero que todo se va a explicar la idea del porque desarrollar este prototipo de terminal de control de producción.

Las empresas que se dedican a la maquila de diferentes productos, por ejemplo camisas, pantalones, componentes electrónicos, etc., necesitan llevar un control estricto en cada sección o mesa de trabajo. En cada sección hay un supervisor que lleva el control con los datos de producción requeridos, por ejemplo cantidad de productos defectuosos y no defectuosos y eficiencia del operario, además de entradas, salidas, cambios de trabajo, comienzos de lote de producción, etc.

Normalmente las ganancias que se obtienen en estas fábricas son relativamente bajas, por lo que para obtener mayores beneficios, se deben disminuir costos, como por ejemplo menor pérdida de material y mayor productividad.

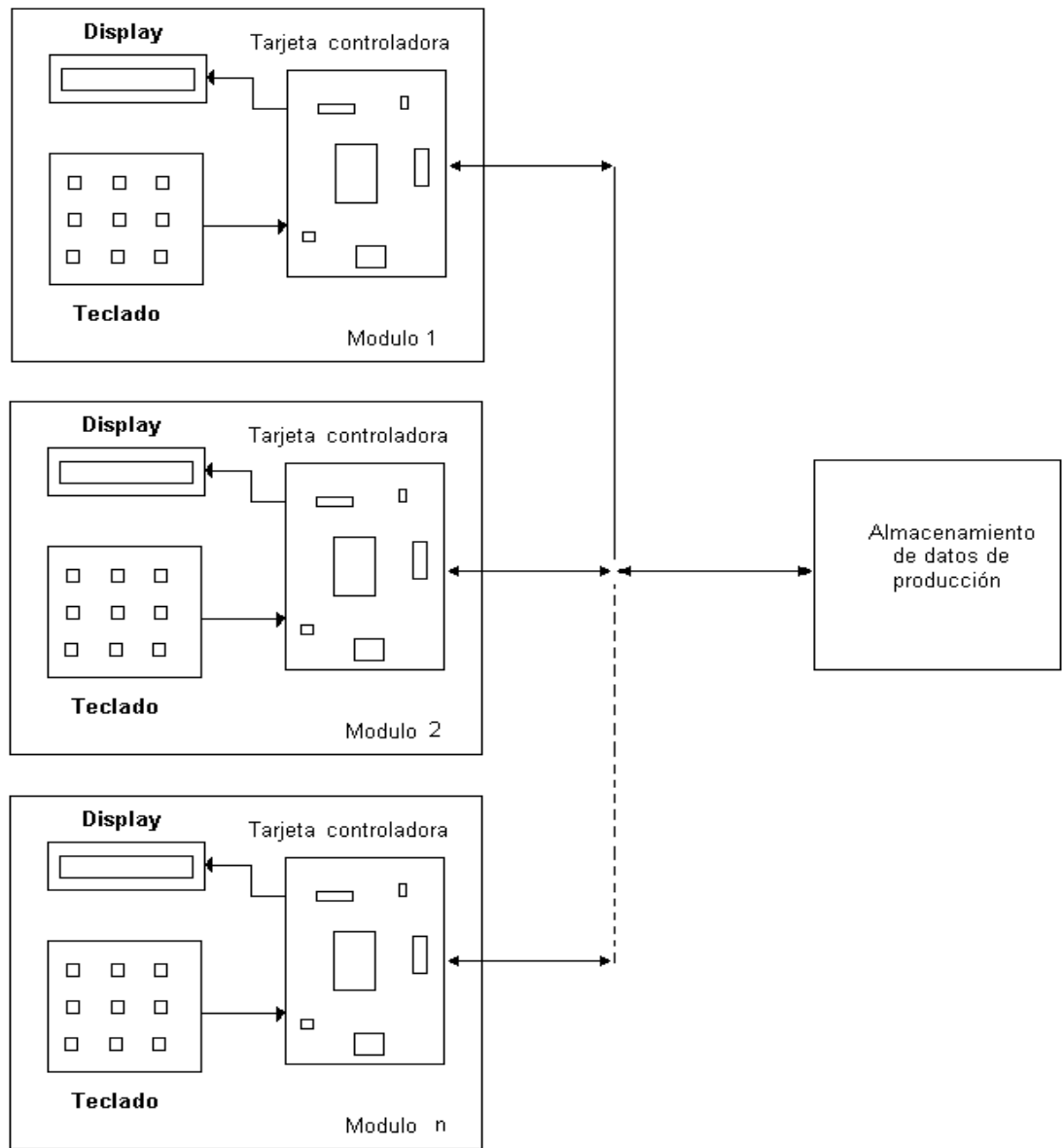
En muchos casos existe un supervisor para cada 30 funcionarios de producción, esto es un costo relativamente alto para la empresa tomando en cuenta que el número de empleados en fábricas de maquila es alto, y por ende tendrían que atender a estos operarios más supervisores.

Los datos que posteriormente tendrán que ser tabulados y accesados al computador para determinar la productividad del día son entre otros: llevar el control de entradas y salidas de los operarios, hora de almuerzo, estimado de producción, faltante para terminar la operación, reducción en el tiempo de marcado lo que aumenta la producción, cambios de trabajo o meta, codificación del lote, la eficiencia del operario durante el día; con respecto al lote llevar el control sobre el número de los procesados, de meta y de faltantes, el estimado a producir al ritmo actual, todo esto con el fin de analizar y tomar decisiones para mejorar la productividad.

La recolección de todos estos datos se vuelve tediosa y el proceso es lento, ya que se realiza manualmente en muchas ocasiones. Este tipo de fábrica necesita un sistema que le permita obtener fácilmente, información actualizada de cada mesa de trabajo en el preciso momento que el producto se va generando.

De acuerdo a esta necesidad el problema consiste en implementar un sistema que realice automáticamente el proceso de recolección de información de cada mesa de trabajo en la fábrica.

Así, se pretende con este sistema, corregir esa deficiencia de recolección de datos, que resulta muy tediosa y lenta para los supervisores con el fin de velar por la producción y el buen rendimiento de la compañía, de tal forma que la empresa corrija inmediatamente la fallas y prevenga futuros problemas. También, es una manera de estudiar el proceso de producción para eventualmente mejorarlo y aumentar la eficiencia de la empresa.



PAINT

Figura 1.1. Diagrama de bloques del proceso a implementar en una fábrica

En la sección de la fábrica donde se va a implementar este sistema, cada mesa o módulo de trabajo va a tener una terminal de control de producción que posteriormente van a ser transmitidos al computador para recopilar los datos y además de mandar informes al operario. El diagrama de bloques de este proceso se muestra en la figura 1.1. Hay que destacar que este último punto se debe realizar un programa que sea amigable con el encargado de recopilar los datos, pero lo que concierne al proyecto de graduación no forma parte, solamente lo correspondiente a la implementación de la terminal, a su respectiva programación y el diseño del módulo de comunicación con la computadora.

A la terminal van a tener acceso varias personas con su respectivo código de barras, el operario, el supervisor y el encargado de control de calidad, este con el fin de verificar los productos defectuosos para posteriormente corregir esos errores de producción.

El problema consiste en que al presentarse algún error en los procedimientos se fabrican partes defectuosas o que no cumplen con alguna especificación del cliente.

Estos errores se encuentran hasta el día siguiente en el departamento de producción cuando revisan las boletas del día anterior, por lo que las acciones correctivas se aplican un día después de haber iniciado el proceso, causando que se pierda la producción de las ocho horas del día, pérdidas tanto de materia prima como pérdidas por el tiempo invertido en ese producto.

2.2 Requerimientos de la empresa

La empresa estableció que los requerimientos para la realización de la terminal del proyecto fueran los siguientes:

Se requiere de una terminal para la recolección de datos en el proceso de fabricación, inicialmente de maquila de ropa, el costo es un factor muy importante.

Se deben controlar diferentes variables, como tiempo, costos, excepciones a operaciones, operaciones a realizar, lotes de producción.

Debe proveerse de un sistema para funcionamiento por niveles de operador (supervisor y operario).

Se desea una comunicación bidireccional entre la terminal y el PC, debido a que no solo se desea mostrar y manipular la información recibida por la terminal para ser mostrada en el PC, sino que se quiere retroalimentar al operario con datos de su productividad y calidad de trabajo.

Microcontrolador: 68HC908GP32, tiene la ventaja que tiene 512 Bytes de RAM, 32kBytes de Flash, además de 33 puertos de I/O.

Pantalla LCD: Se utiliza una de 4 líneas por 20 caracteres para mostrar la información de la terminal. Debe operar en temperaturas de trabajo normales (15 a 40°), debe ser fácil de leer y además de mostrar el máximo de información posible, sin sobrecargar la lectura del operario, esto es que en la pantalla esté estrictamente lo relevante a su línea de producción y no vaya a confundir al operador de la terminal.

Teclado: Matricial de 16 teclas, dígitos del 0-9, Enter, Cancel, Clear y tres teclas de función que combinadas con los números dan 27 funciones adicionales.

Sensores: Se utilizan sensores lineales tipo CCD para hacer la lectura de códigos de barras debido a su simplicidad, resolución y costo.

Fuentes de alimentación: El circuito en un principio se compone de dos fuentes de alimentación, tanto para las operaciones del microcontrolador, como para el display y además para el juego de leds que servirán como un sistema lumínico para la parte de los sensores.

Estas fuentes son de 5v que básicamente están formadas por medio de un regulador de voltaje, el cual combinando algunos valores de resistencias y capacitores a la entrada y salida del mismo se obtiene la tensión de salida requerida, que ni siquiera vaya a perjudicar significativamente el valor del voltaje de la misma cuando alimente a los diferentes componentes electrónicos, específicamente que cuando esté con carga, la tensión no se vea reducida.

2.3 Solución propuesta

La empresa SMARTECH tenía proyectada la solución, la cual consiste en desarrollar un prototipo terminal de control de producción, basado en un microcontrolador de la familia HC08 de Motorola.

La terminal se conforma principalmente del microcontrolador 68HC908GP32, una pantalla (display), un teclado, sensores CCD lineales y un puerto de comunicaciones RS-485 como se muestra en la figura N°2.1.

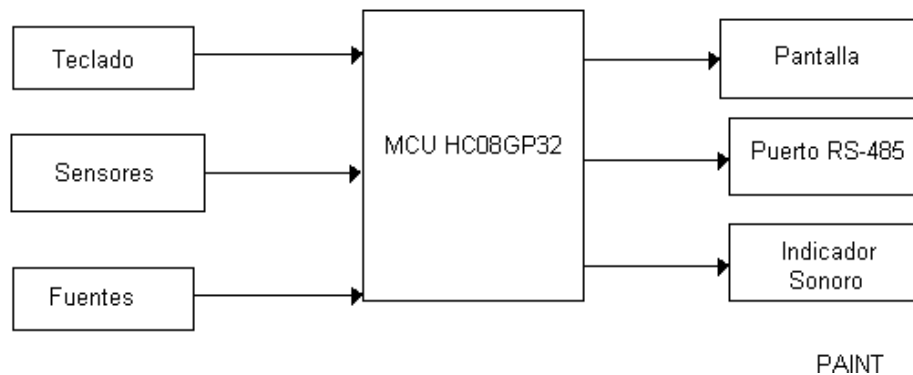


Figura 2.1 Diagrama de bloques general de la terminal.

El funcionamiento de esta terminal de control de producción es básicamente para llevar la productividad del operario durante el día. Cada terminal va a ser manipulado por el supervisor, el operario y además el control de calidad.

Cada uno de ellos va a tener un código de identificación mediante una tarjeta con el código de barras, los sensores se encargarán de leer este número y será enviado al microcontrolador para que éste lo procese.

El supervisor o encargado de la sección de trabajo interacciona con el módulo introduciendo los datos requeridos sobre la producción mediante el teclado y el display, después de haberse identificado con su respectiva tarjeta. Él por ejemplo puede determinar la productividad que requiere que cierto operario realice durante el día, ya que el módulo podrá determinar el rendimiento por medio de las cantidades que se hayan producido y por medio del rendimiento del día que se había propuesto. También puede cambiar la eficiencia, se programa también la hora de inicio y fin del almuerzo. Todas estas variantes de control de producción se pueden realizar también desde la computadora.

El operario tendrá un código de identificación para marcar su hora de entrada y de salida, si se hizo un cambio de trabajo o de lote de producción.

El supervisor del control de calidad también contará con un acceso a la terminal, para verificar ciertos aspectos de producción más que todo sobre las fallas en el producto.

CAPÍTULO 3

PROCEDIMIENTO METODOLÓGICO

3.1 Metodologías relacionadas con el MCU:

3.1.1. Conocer el funcionamiento del microcontrolador de Motorola 68HC908GP32.

Por medio del manual del GP32, se investigó su funcionamiento y módulos de operación que permiten hacer más fácil el prototipo a implementar.

3.1.2. Investigar el funcionamiento de las tarjetas con códigos de barra.

Se recurrió a informes con proyectos similares que trabajaron con código de barras, guías y por medio de Internet, este con el fin de adecuar los sensores.

3.1.3. Conocer el programa de programación y emulación de Motorola WinIDE.

Con el respectivo manual de programación del microcontrolador GP32, se estudió el programa, para el ensamblador y el emulador del respectivo micro.

3.1.4. Conocer el funcionamiento del hardware del emulador la familia HC08 de Motorola

Por medio del manual del WinIDE, se investigó el funcionamiento del emulador para ver cómo interactúa con el computador.

3.1.5. Hacer pruebas con el emulador de Motorola.

Teniendo listo la programación del GP32, se procedió con el emulador hacer las pruebas respectivas con el hardware (teclado y display) para verificar su funcionamiento.

3.2. Metodologías relacionadas con el diseño de la terminal:

3.2.1. Estudiar el diseño actual del circuito del prototipo a implementar.

Con el diseño implementado por parte de la empresa, se analizó para hacerle modificaciones, aparte de hacer sugerencias, para posteriormente montarlo en la placa de pruebas.

3.2.2. Conocer el funcionamiento de los circuitos especiales:

3.2.2.1. Conocer el funcionamiento de los sensores ópticos tipo CCD lineal.

Se investigó mediante, manuales e Internet, los tipos de sensores ópticos CCD para escoger el adecuado o dejar los que ya están. Además se hicieron pruebas con el osciloscopio para escoger el más adecuado.

3.2.2.2. Circuito integrado de comunicación RS-485.

Mediante bibliografía especializada sobre protocolos de comunicación, hojas de datos técnicos e internet se investigó sobre este tipo de comunicación y se comparó con otras.

3.2.2.3. Pantalla de cristal liquido.

Se investigó en manuales de fabricación, e internet el funcionamiento de este dispositivo electrónico.

3.2.3. Implementar el hardware del prototipo terminal de control de producción.

Teniendo hechas todas las modificaciones del circuito se procedió a montar el prototipo en placas de prueba tipo wire-wrap, el microcontrolador va colocado en una base especial, la cual no se encontró en el país y en el exterior resultó ser muy cara, entonces se procedió a traer una tarjeta que tiene el microcontrolador incorporado, se hicieron modificaciones a la placa de los sensores, display y teclado para que se adaptara a esta tarjeta.

3.3. Metodologías relacionadas con el software:

3.3.1. Implementar el software del microcontrolador de Motorola 68HC908GP32 para cumplir con los requerimientos de producción.

Después de haber conocido el funcionamiento del microcontrolador tanto el hardware como el software, además del emulador se procedió a realizar la programación del mismo en el WinIDE, tanto para la pantalla, teclado, conversión de códigos de barras y comunicación serie con la PC.

CAPÍTULO 4

DESCRIPCIÓN DEL HARDWARE UTILIZADO

A continuación se detallan los dispositivos que se utilizan para la elaboración del proyecto, estos fueron seleccionados con los criterios explicados en la sección 3 de este documento.

Se utiliza un teclado matricial de 16 teclas, es decir, el teclado tiene 8 pines como se ve en la figura 4.1, cada una de las teclas es una combinación de pines que el microcontrolador va a estar revisando constantemente cuando una tecla esté presionada.

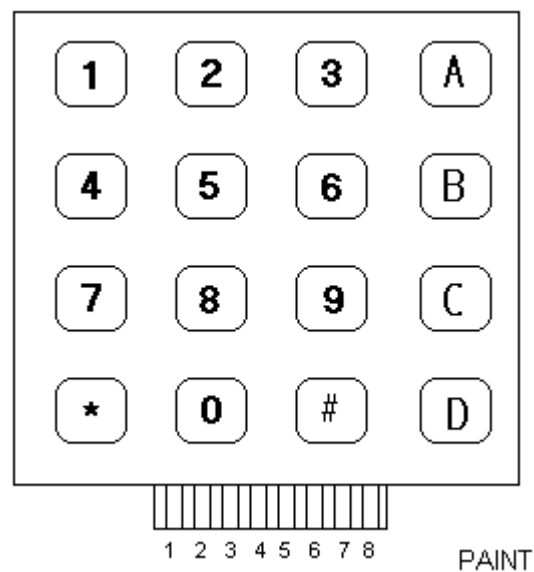


Figura 4.1 Diagrama de bloques del teclado utilizado

El diagrama matricial se muestra en la figura 4.2, el cual los pines de salida corresponde a la tabla 4.1

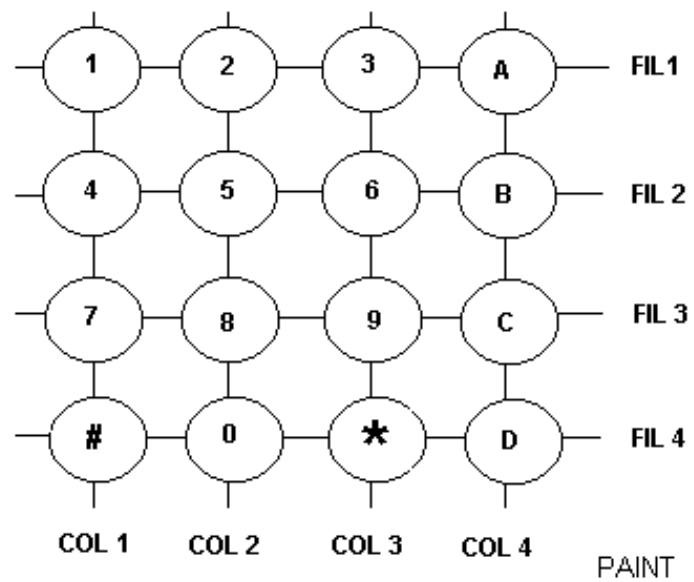


Figura 4.2 Diagrama matricial del teclado utilizado

Tabla 4.1 Definición de los pines de salida del teclado

Salida Pin Número	Símbolo
1	FIL 1
2	FIL 2
3	FIL 3
4	FIL 4
5	COL 1
6	COL 2
7	COL 3
8	COL4

Se utiliza una pantalla de 20x4 líneas, es decir 20 líneas de ancho por cuatro de alto, tiene un controlador integrado y para manejarla se utilizan tres líneas de

control y ocho líneas que corresponden a datos. El bloque funcional se muestra en la figura 4.3.

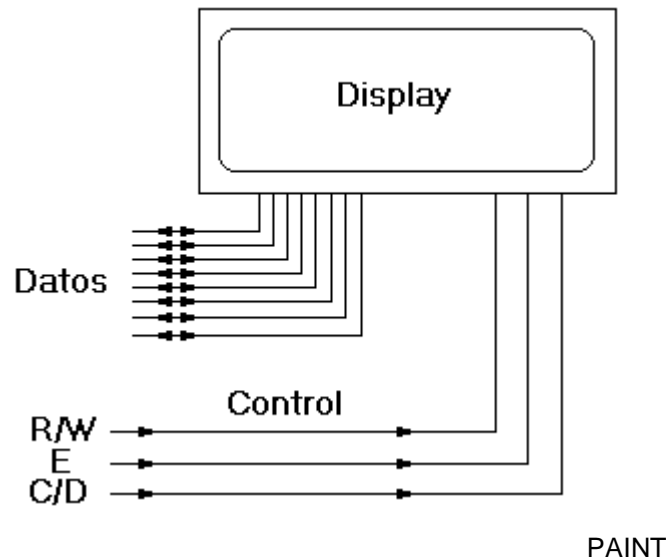


Figura 4.3 Bloque funcional de la pantalla

El controlador del display puede ejecutar órdenes como inicializar display, limpiar pantalla, mover el cursor, leer de display, escribir un dato y pasar de línea entre otras funciones, se puede encontrar una descripción de las instrucciones utilizadas en el proyecto en el apéndice 1.

Para manejar las 8 entradas del teclado, las once entradas de la pantalla y las siete entradas para los sensores, se utiliza un microcontrolador MC68HC908GP32 que además tiene un puerto serie incorporado,

Las principales características físicas y eléctricas de este microcontrolador se pueden encontrar en el manual del HC08GP32 o en la dirección web: <http://www.motorola.com/sps> anexo 1, entre otras características podemos citar que tiene modos de operación a 3 y 5V, modo de bajo consumo de potencia y una línea de interrupción externa que es enmascarable.

Básicamente este microcontrolador tiene 512 bytes de RAM, 32Kbytes de FLASH, tiene cinco puertos y varios módulos dentro de los que podemos citar:

a) Break module: Este módulo produce una interrupción cuando el puntero de instrucción del CPU del microcontrolador llega a un punto determinado del programa, es similar a poner un “Break Point” en algún programa cuando se ejecuta la simulación paso a paso en una PC.

b) Clock Generator Module: Se encarga de generar un reloj interno para que el CPU funcione, además de generar relojes para la UART¹ y para el PLL².

c) Keyboard Interrupt Module : Es otra opción para manejar teclados con el puerto A, en este modo se atiende el teclado por medio de una interrupción.

d) Low Voltaje Inhibit :Causa que el CPU se detenga y se inhiban las salidas cuando el voltaje de alimentación sea menor de 3V en modo de operación a 5V, el problema que se presenta si la tensión es tan baja es que la memoria RAM se pierde, por lo que las variables del programa se verían afectadas causando salidas de control no deseadas, el LVI se puede programar para que produzca un reset al detectar un voltaje menor que el deseado.

e) Monitor ROM : Cada microcontrolador tiene un espacio de memoria llamado Monitor ROM que es donde están guardadas las rutinas que el software de la PC usa para programar los microcontroladores, estas rutinas se pueden utilizar también en cualquier programa para cambiar parámetros del mismo reprogramando la FLASH donde se encuentra el código del programa, para entrar en este modo monitor se puede hacer conectando la línea de interrupción a un voltaje mayor a 1.5 veces el voltaje de alimentación (7.5 V en este caso para entrar) o entra directamente a este modo si el vector de reset se encuentra borrado.

¹ Unidad asincrónica de transmisión y recepción

² Phase Locked Loop : Circuito de control utilizado en algunos sintonizadores de FM y que consiste en enfazar dos señales para asegurar la correcta recepción de la señal de información que se quiere.

f) Serial Communications Interface Module : Es el modulo del puerto serie, en este se puede programar el largo de las palabras de 8 ó 9 bits, bits de inicio y bits de parada así como programar la velocidad de transmisión y recepción por medio de divisores de frecuencia que se describen en sección 18 del manual del microcontrolador en la dirección web: <http://www.motorola.com/sps> .

El GP32 tiene otro módulo para comunicaciones serie (serial peripheral interface module SPI) que es mejor para redes de microcontroladores porque tiene modo de esclavo y maestro así como dos registros independientes de entrada y salida que permiten una comunicación full duplex, (el SCI tiene un solo registro de datos), sin embargo se utiliza el SCI porque es más sencilla su programación y queda como opción a futuro utilizar el SPI.

El microcontrolador tiene los siguientes puertos:

Tabla 4.2 Puertos del microcontrolador MC68HC908GP32

Puerto	Características
A	8 bits bidireccionales de propósito general con interrupción para teclado
B	8 bits bidireccionales de propósito general, además cada pin puede ser utilizado como entrada analógica ya que posee un convertidor analógico – digital multiplexado
C	5 bits bidireccionales de propósito general
D	5 bits bidireccionales de propósito general
E	Puerto serie, posee los pines Tx y Rx

Se probaron diversos sensores “Charge Coupled Device” (CCD) y con LED's con diferentes tipos de colores y ángulos de luminosidad

Uno de ellos consiste en un arreglo lineal de 64 x 1 fotodiodos, asociado con un amplificador de carga y un retenedor de píxeles que provee los tiempos de inicio y

de final para la lectura de los mismos, este tiempo se denomina tiempo de integración.

Las medidas de cada píxel es de 120 μm de alto por 70 μm de ancho. La operación de este sensor es por medio de una lógica de control interna que requiere sólo una entrada serial y un reloj.

Los otros dos sensores tienen la misma función, la diferencia reside en la lectura de píxeles, uno es un arreglo de 128 x 1 fotodiodos, las medidas para este caso es de 63.5 μm de alto por 55.5 μm de ancho. Otro sensor es más grande porque consiste de dos secciones de 128 fotodiodos cada una, lo que se obtiene un arreglo de 256 x 1; es más efectivo pero de mayor precio también.

El integrado a utilizar para la comunicación es el LTC1487, el cual es un “diferential transceiver”³ de baja potencia y alta impedancia que soporta una transmisión de hasta 256 cargas. Una descripción más detallada se encuentra en el anexo 1.

³ Tranceiver: Dispositivo que habilita la transmisión y recepción de datos

CAPÍTULO 5

DESCRIPCIÓN DEL SOFTWARE UTILIZADO

El programa del microcontrolador se puede escribir en cualquier editor de texto y luego compilarlo en un programa que traduce el código a lenguaje de máquina.

El código que se utiliza en este programa es el ensamblador de PEMicro que está incluido con el software WINIDE Developement Kit y que está basado en el set de instrucciones para la familia HC08 que se encuentra en la sección 10 del manual del GP32 que se puede obtener en la dirección: <http://www.motorola.com/sps>. El programa se debe hacer utilizando únicamente este set de instrucciones y no soporta el uso de lenguajes de alto nivel cuando se compila con PEMicro.

También se puede hacer el programa utilizando código del lenguaje C si se utiliza el MCUEz para compilar o se puede programar en el lenguaje C si se utiliza el COSMIC, sin embargo no está disponible en la empresa.

El proyecto se realizó utilizando el WINIDE Developement Kit, la interface del WINIDE se puede encontrar en el apéndice 3, presenta opciones de compilación, simulación y programación, como todos los microcontroladores tienen el Monitor ROM este software se puede utilizar en un programador serie que se puede fabricar fácilmente y cuyo circuito y explicación se encuentran en la sección 15 del manual.

El software del sistema de adquisición de datos realiza por el momento la siguiente secuencia de acciones:

Inicializa la pantalla, la limpia y muestra un mensaje de inicio: en este momento el usuario debe presionar la tecla respectiva para el menú que quiera desplegar en pantalla, el diagrama de flujo se muestra en el figura 6.2.

Para escribir en la pantalla o se utiliza una rutina que lea una tecla, la muestra en pantalla, guarda en memoria y espera a que el usuario deje de presionar la tecla para evitar repeticiones, el sistema detecta si la tecla presionada es un número o una letra, esta se muestra en la figura 6.3.

CAPÍTULO 6

ANÁLISIS Y RESULTADOS

6.1 Explicación del diseño

El prototipo que se construyó posee todas las características que se habían planteado inicialmente, como lo son el ingreso de datos desde teclado y presentación de los datos en un display.

El display de cristal líquido, es controlado por el microcontrolador por medio de todo el puerto D para los datos y tres pines del puerto C para el control de la pantalla.

El teclado está en todo el puerto A del microcontrolador, la transmisión y recepción de la comunicación serie están a cargo de los dos pines del puerto E y la manipulación de los 5 sensores CCD está a cargo del puerto B. Cinco pines se utilizan para la entrada serial de cada CCD y un pin para el reloj.

Si se desea, se puede cambiar a un microcontrolador MC68HC908GP20 que es idéntico físicamente al GP32 utilizado pero tiene la diferencia que posee 20 Kb de memoria FLASH EEPROM (el GP32 tiene 32 Kb) solo debe cambiarse el vector de inicio en el programa del dispositivo cambiando en la línea de inicio de la Flash: el 8000h por un B000h.

Con respecto al diseño de las tarjetas se escogió el formato ITF, este es un formato con alta densidad de datos y ofrece las siguientes características:

- El tamaño de la etiqueta puede ser más pequeño que otros códigos con los mismos dígitos. Esto es efectivo al imprimir códigos de barras en espacios reducidos.
- Más datos pueden ser contenidos comparados con otros códigos, si el tamaño de la etiqueta es la misma.

- Son posibles barras más anchas, si el tamaño de la etiqueta y el número de dígitos es el mismo. Entre más gruesas sean las barras la lectura del código es mejor.

En la figura 6.1 se muestra una tarjeta con el formato establecido.



Figura 6.1. Formato ITF para las tarjetas

En este caso se muestran 5 juegos de barras, uno por cada CCD, es por eso la separación entre ellos cada juego de estos corresponde a dos números.

El ancho de las barras se calculó para una mejor adaptación a los sensores y además hay una norma que dice que una barra gruesa equivale a 2.5 de las delgadas.

A continuación se presentan las subrutinas que se utilizan en la terminal:

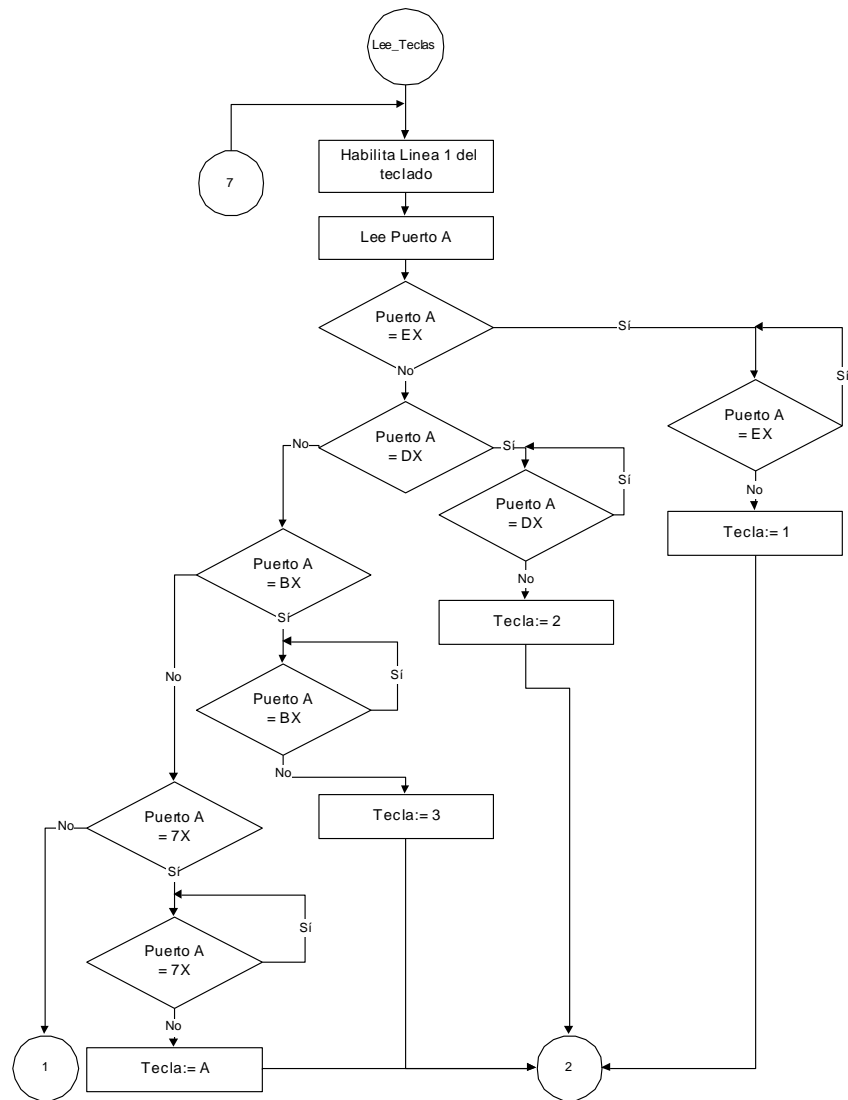
Subrutinas de teclado

a- Subrutina que lee las teclas del teclado:

Esta subrutina lee vía polling los pines del puerto A, se activa el registro de los pull ups del puerto A con unos, luego se define el puerto A como entradas en primera instancia, posteriormente se definen unos en la parte baja del puerto A

(uno por uno) cuando se presiona una tecla en la entrada respectiva se pone un cero dando por leída esa tecla.

El algoritmo que describe a esta subrutina se muestra en el siguiente diagrama de flujo:



VISIO

Figura 6.2. Diagrama de flujo de lectura del teclado

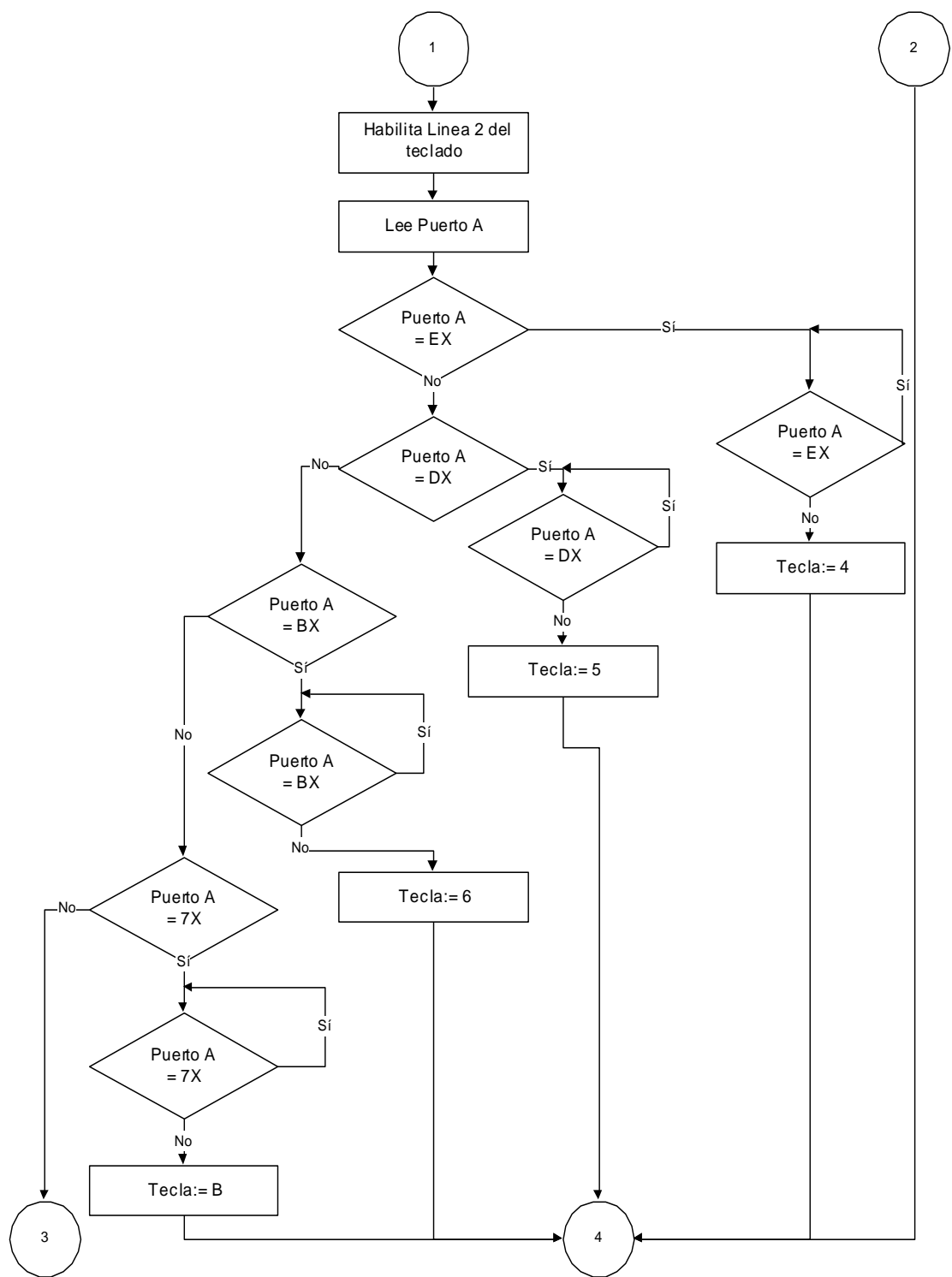


Figura 6.3. Continuación diagrama de flujo de lectura del teclado

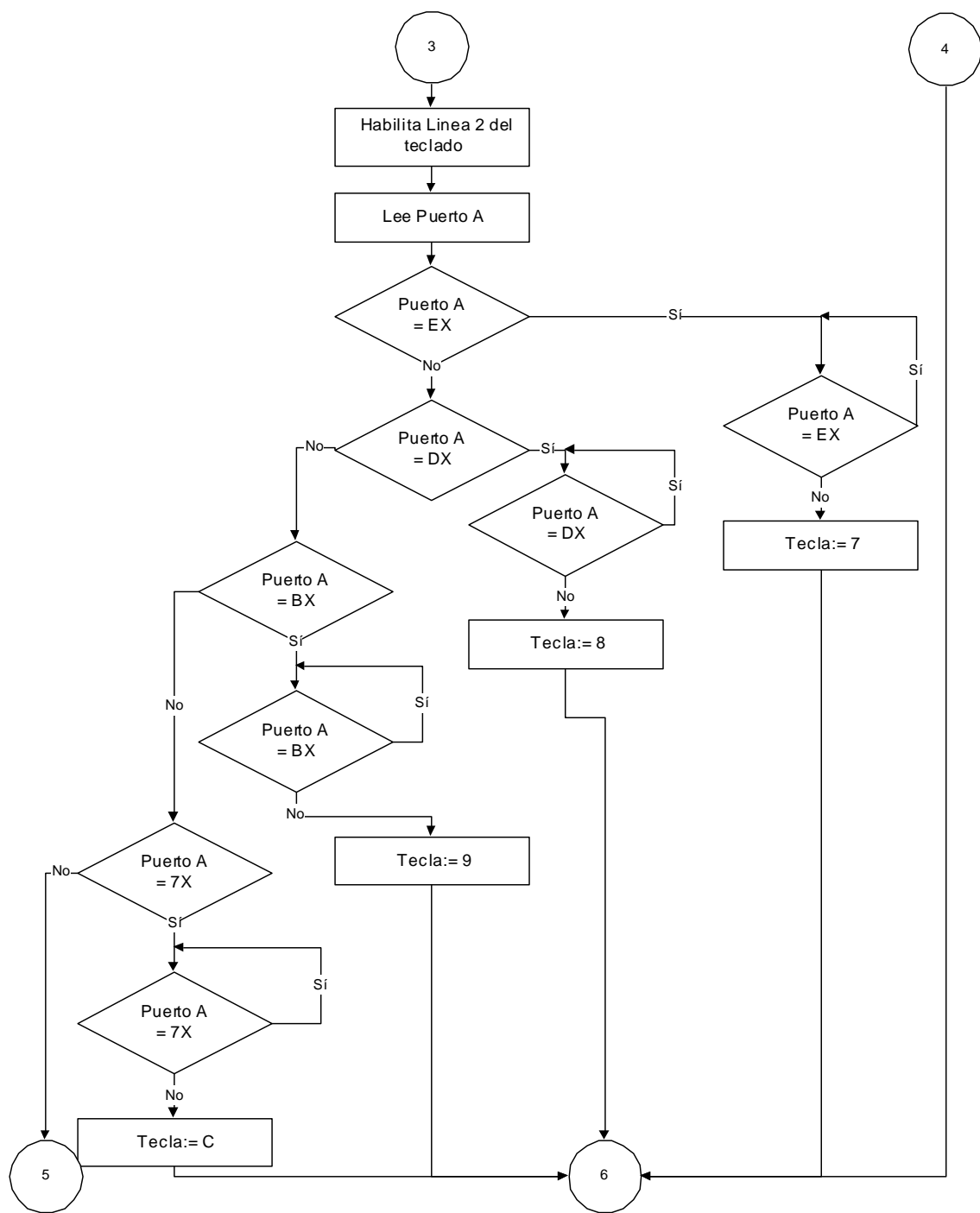
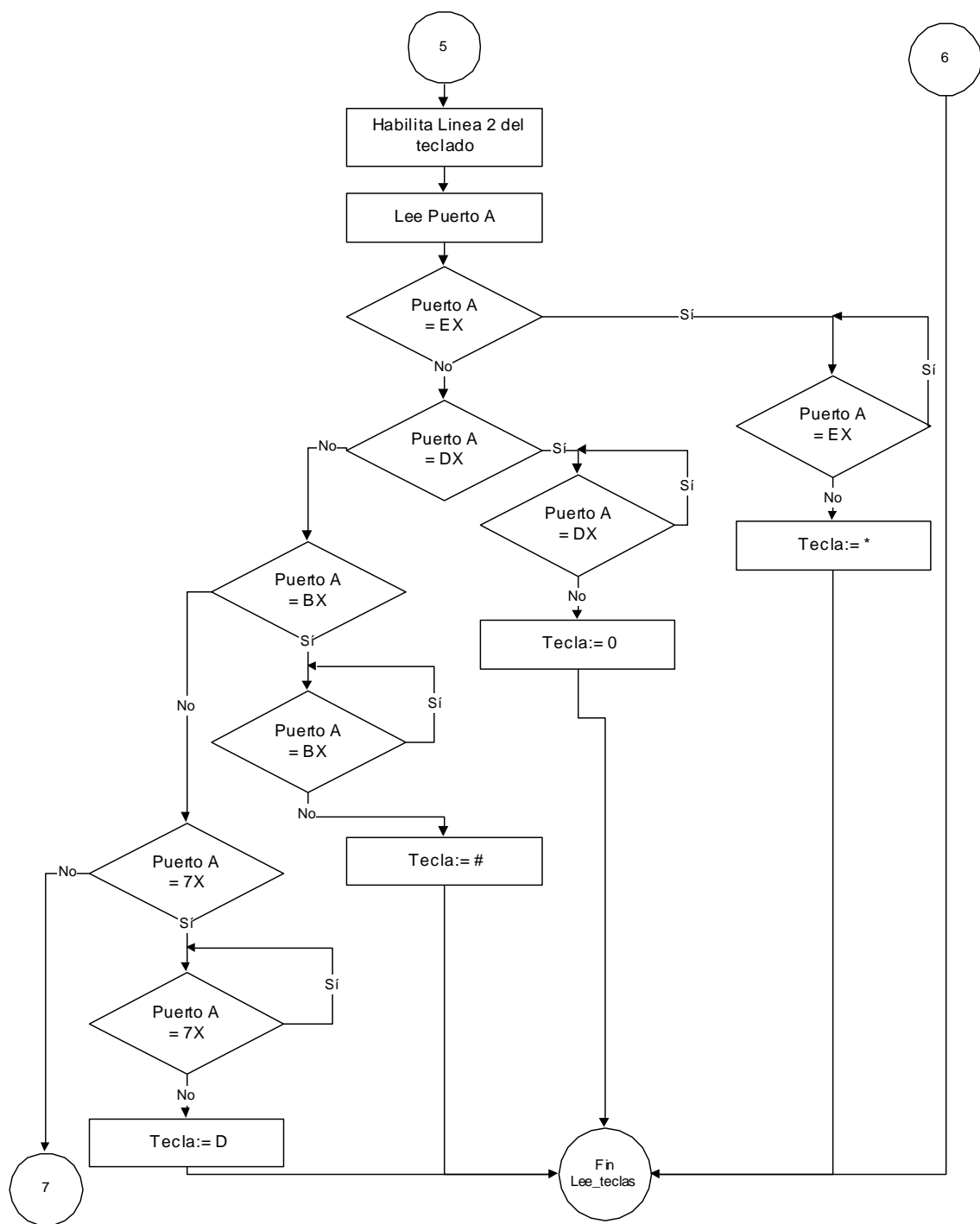


Figura 6.4. Continuación diagrama de flujo de lectura del teclado



VISIO

Figura 6.5. Continuación diagrama de flujo de lectura del teclado

b - Subrutina que escribe el valor de las teclas:

Realiza la misma rutina de lectura, solo que adicionalmente envía el valor de la tecla respectiva al puerto D, que es del display.

Esta subrutina tiene un retardo de 10 ms para evitar rebotes, cuando se haya leído la tecla respectiva salta a una etiqueta que espera 10ms, luego vuelve a verificar si todavía está presionada, si lo está salta a otra etiqueta que se queda ciclada hasta que se deje de presionar la tecla; si no vuelve al inicio a leer todas las teclas.

El algoritmo de esta subrutina es muy similar al que lee teclas, con ciertas inclusiones, a continuación se muestra el diagrama de flujo.

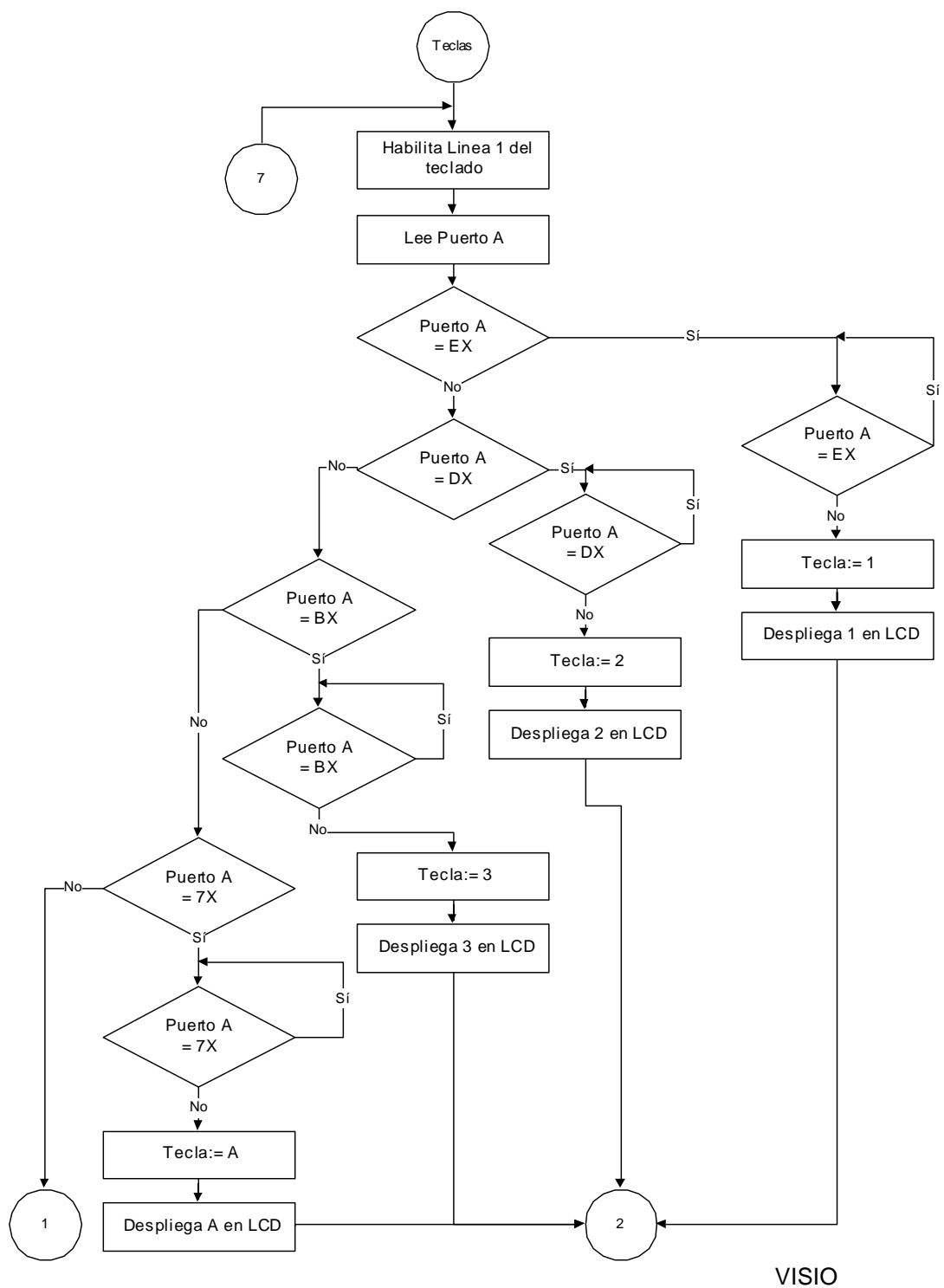


Figura 6.6. Diagrama de flujo de escritura de las teclas

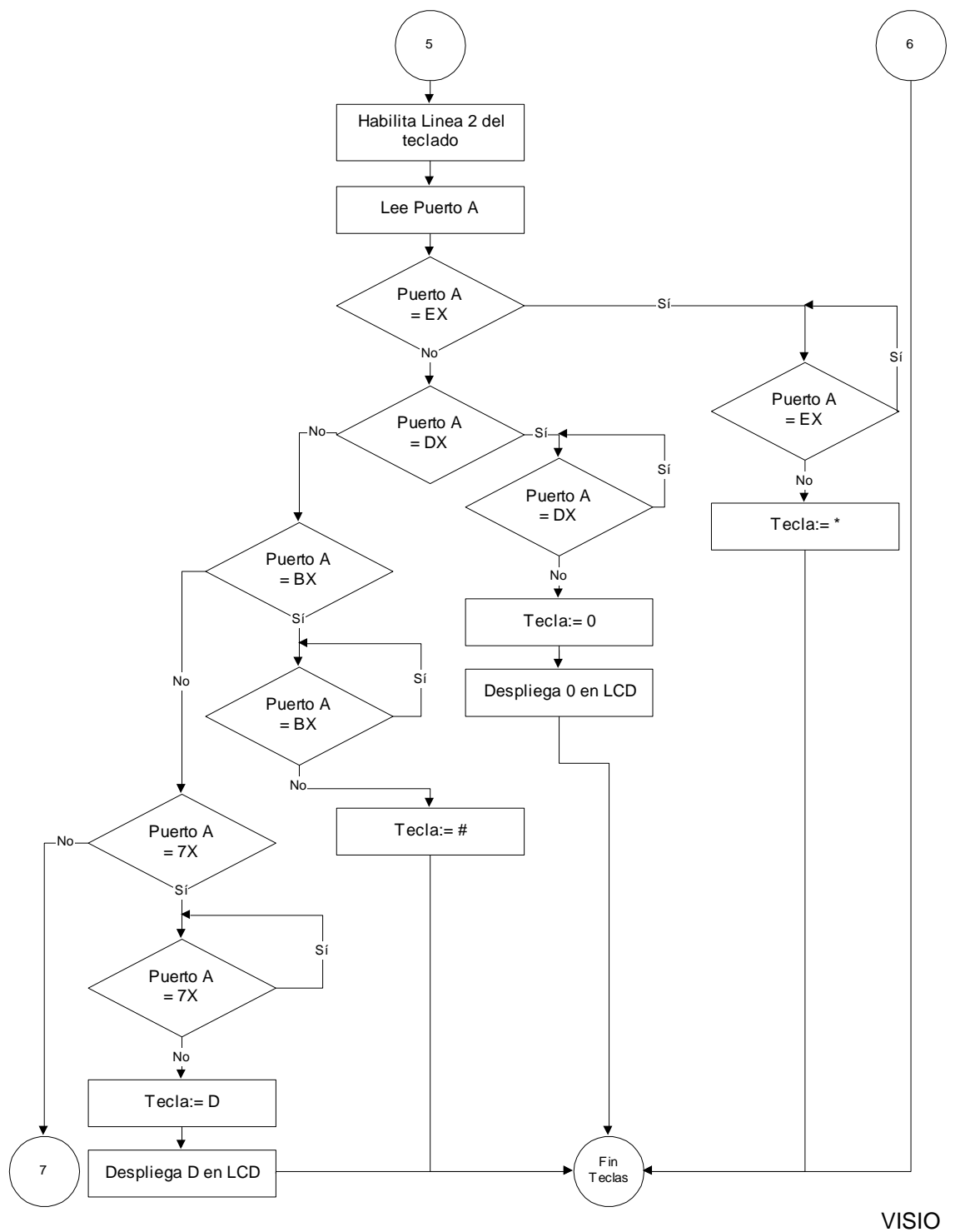


Figura 6.7. Continuación diagrama de flujo de lectura del teclado

Subrutinas de la pantalla

a- Subrutina de inicialización del display

Después de haber inicializado el puerto D como salidas se procede a realizar la respectiva inicialización del display.

La inicialización consiste en enviar códigos de instrucción al puerto C donde se encuentra los pines de control del display.

b- Subrutina de instrucción

Esta subrutina da la respectiva instrucción al display vía puerto C que es el puerto de control de la pantalla.

Consiste en enviar unos y ceros a los pines 0, 1 y 2 del puerto C que identifica la instrucción en el display.

Por ejemplo la posición en cada una de las líneas del mismo, el inicio de la línea uno corresponde a la instrucción 80H, la de la línea 2 es C0H, la de la línea 3 es 94H y la de la línea 4 es la D4H.

c- Subrutina de dato.

Cuando se haya dado la respectiva instrucción se procede a escribir el dato enviado al puerto D por medio del puerto de control del display C.

Es el mismo caso con la subrutina de instrucción, solamente que cuando se manda un dato al puerto D, que son los pines de datos de los del display llama a esta subrutina, haciendo prácticamente lo mismo que en instrucción.

Subrutinas para desplegar mensajes

a- Carga mensaje:

Carga la dirección del mensaje que se desea desplegar en el puntero mensaje.

Los mensajes son previamente cargados en la memoria en su dirección respectiva, por medio de la directiva include, los mensajes están en una tabla en mensajes.inc.

Inicialmente se carga el mensaje con su tamaño respectivo y en la línea donde se va a desplegar, ya realizado esto pasa a la siguiente rutina que se encarga de desplegar mensaje seleccionado.

b- Despliega mensaje:

Se encarga de desplegar los datos apuntados por la variable mensaje en la posición del display especificada.

Las variables que son utilizadas en esta subrutina para que se maneje el despliegue en la pantalla, son líneas y posición

El puntero también va a la posición de memoria donde se encuentran los diferentes idiomas según el idioma seleccionado, que está en una posición específica de memoria.

Subrutinas de lectura de código de barras

a- Lectura píxeles:

Se encarga de leer todos los píxeles del sensor CCD. Esta subrutina envía las señales de entrada serial y CLK al sensor, necesarias para poder hacer la lectura. Se encarga hacer la conversión analógica – digital y almacenar el valor convertido en la memoria RAM, a partir de la dirección \$0040 hasta la dirección \$007F.

b- Conversión:

Se encarga de convertir los píxeles al dígito representado en el código de barras. Para hacer la conversión se implementaron 4 subrutinas. La primera, `pixeles_a_bits`, se encarga de convertir cada píxel leído en un 1 si es mayor que el valor medio de los píxeles o a un 0 en caso contrario. Estos nuevos 64 datos (1 bit c/u) se almacenan en 8 variables de 1 byte. La siguiente subrutina, `bits_a_barras` se encarga de interpretar el ancho de cada barra, esto se logra contando cuantos bits iguales se encuentran juntos. La tercera subrutina, `barras_a_numeros` se encarga de interpretar el código de acuerdo al ancho de las barras leídas. La cuarta subrutina, `conversión`, es la principal, se encarga de llamar las anteriores para obtener la lectura final. Primero obtiene el valor medio de los píxeles, llama a `pixeles_a_bits`, luego convierte los bits a barras llamando a `bits_a_barras`, luego se encarga de interpretar el numero de barras convertidas leídas y los anchos para determinar si se debe cambiar el valor medio, en cuyo caso lo modifica y vuelve a llamar las dos subrutinas anteriores y así sucesivamente hasta que los anchos de las barras obtenidos lleven a una lectura valida o después de varios intentos, en cuyo caso la lectura no fue posible.

Cuando se obtienen anchos de barras que puedan generar una lectura valida se llama a la subrutina `barras_a_numeros` y se obtiene el código representado por las barras en el caso de existir, si no es valida, devuelve un valor conocido que no representa un numero decimal.

Píxeles a bits:

Esta subrutina convierte el valor de cada píxel en un bit, dependiendo si es mayor que el valor medio.

Devuelve 64 bits divididos en 8 bytes donde cada bit representa la lectura de cada uno de los píxeles.

Bits a barras:

Toma los 8 bytes que se obtuvieron de la subrutina pasada y obtiene el número de bits consecutivos que tienen el mismo valor para así determinar el ancho de cada barra. Como salida da el tamaño de cada una de las 10 barras.

Barras a números :

Toma los datos de la subrutina anterior es decir el ancho de cada barra, entonces toma el ancho de cada una y lo compara con el número mínimo de píxeles que el sistema leería para una barra ancha. Si la barra tiene ese número mínimo o más píxeles asigna un 1 al bit correspondiente a la variable que va a representar el número, y si es menor asigna un cero y se representa como una barra delgada, luego toma esta variable y la compara con la tabla de constantes que son los valores de los códigos de barras según el formato ITF.

Subrutina de borrar memoria y grabar en FLASH:

Estas subrutinas fueron tomadas del manual del microcontrolador GP32 las cuales se muestran detalladamente en la sección 11 del manual; pero básicamente consiste en ir programando los registros internos de la memoria FLASH cada cierto lapso de tiempo tanto para programar como para borrar cierta cantidad de bytes.

Subrutina de comunicación:

Recibe: se activa por interrupción cuando ha llegado un byte, entonces almacena el byte en un arreglo hasta recibir el bloque completo.

Envía: envía por el puerto de comunicación una variable que se carga con anterioridad con el dato a mandar.

6.2 Alcances y limitaciones

Los alcances de este proyecto, están basados en la lectura del código de barras, que es una de las partes de mayor complejidad ya que se ven involucradas muchas variables como por ejemplo el tiempo de integración, la definición de la impresión y el tamaño correcto de las barras.

Otro alcance importante es que ese código convertido, es leído por la PC vía puerto serie vía puerto RS 232.

También se logró implementar una subrutina que permite seleccionar diferentes idiomas, sin que se necesite volver a reprogramar el microcontrolador.

El indicador sonoro funciona con la lectura correcta del código.

Con respecto a las limitantes del proyecto:

1- Es que no se cuenta por el momento con terminales para RS-485, por lo cual la comunicación vía ese puerto no se ha podido probar.

2- La terminal sólo lee un tipo de código de barras, el ITF y espaciados específicamente para la adecuada lectura de los sensores.

CAPÍTULO 7

CONCLUSIONES Y RECOMENDACIONES

CONCLUSIONES

1. Se determinó que el mejor formato para códigos de barras es el ITF, principalmente por su tamaño compacto.
2. Se comprobó que los leds con los que se obtuvieron mejores lecturas fueron los de color verde.
3. Se seleccionó el sensor CCD de 64 pixels, por sus características, como su bajo costo y buena señal de salida.
4. Tomando en cuenta el área que cubre cada CCD, las normas de códigos de barras y los resultados obtenidos, se determinó que el ancho de las barras delgadas fueran de 0.3mm y el ancho de las barras gruesas de 0.8mm.
5. Se determinó que para este tipo de aplicaciones, el controlador 68HC908GP32 es óptimo debido a la gran cantidad de puertos I/O, además de los diferentes módulos que posee, como el convertidor analógico digital y la interfase de comunicación serial.

RECOMENDACIONES

1. Se recomienda hacer la lectura de código con un solo sensor CCD, esto reduciría significativamente el costo, no sólo porque economizarían 4 CCD, sino también por la reducción en los leds. Además se podrían leer distintos formatos de códigos de barras.
2. Se recomienda hacer una tarjeta adicional, para cambiar la programación, ya que por medio del Monitor ROM, se utilizan componentes adicionales, que al ponerlos en cada terminal, elevaría el costo, en cambio con una tarjeta diseñada para la programación se utilizarían para las demás terminales.
3. Es mejor utilizar el método de polling donde la computadora personal pregunta a cada dispositivo por los datos que tiene en su memoria que permitir a cada dispositivo enviar datos en cualquier momento, esto es para evitar que dos dispositivos envíen datos al mismo tiempo.

BIBLIOGRAFÍA

Manual “HC08 Advance Information”, Motorola, 1999

Dirección de Motorola en Internet:

<http://www.motorola.com/sps>

Dirección de PEMicro en Internet:

<http://www.pemicro.com>

Dirección de Linear Technology en Internet:

<http://www.linear-tech.com>

APÉNDICES

Apéndice 1: Resumen de las instrucciones utilizadas en el programa














TABLA A1 Resumen de las instrucciones utilizadas en el programa

CLRA	CLEAR A	Asigna 00 al registro A que corresponde al acumulador del microcontrolador
CLR X	CLEAR X	Asigna 00 al registro X que corresponden a los bits más significativos del acumulador, además sirve como registro índice para el modo de direccionamiento indexado
MOV #\$FF,DDRA	MOVE	Mueve un número FF a la dirección DDRA (Data Direction Register A) ⁴
JSR Subrutina	Jump subroutine	Salta a la subrutina especificada
BRCLR \$0,PORTA,Etiqueta	Branch if clear	Salta a la etiqueta si el pin 0 del puerto A está en cero lógico
BRSET \$0,PORTA,Etiqueta	Branch if set	Salta a la etiqueta si el pin 0 del puerto A está en uno lógico
LDA #\$FF	Load A	Carga FF en el registro A
LDHX #\$FF	Load H:X	Carga FF en los registros H:X
STA Variable,X	Store A	Guarda el contenido del registro A en el espacio de memoria apuntado por variable:X , donde X es el registro de desplazamiento
INC Variable	Increment	Incrementa la variable
INCA	Increment A	Incrementa registro A
INCX	Increment X	Incrementa registro X
JMP Etiqueta	Jump	Salta a la etiqueta
CMP Variable	Compare	Compara la variable con el registro A, almacena el

⁴ Indica si el puerto es de entrada o salida, un 1 indica pin de salida y un 0 indica pin de entrada

		resultado en A y modifica las banderas del microcontrolador
BEQ Etiqueta	Branch if equal	Salta a la etiqueta si la bandera de igual está activada
DBNZ Variable,Etiqueta	Decrement and branch if not zero	Decrementa la variable y salta a la etiqueta si la bandera de cero no se activa
AND Variable	AND A	Ejecuta la operación lógica correspondiente a una compuerta AND de la variable con el acumulador

Apéndice 2: Funciones de las teclas que presenta el WinIDE Development kit

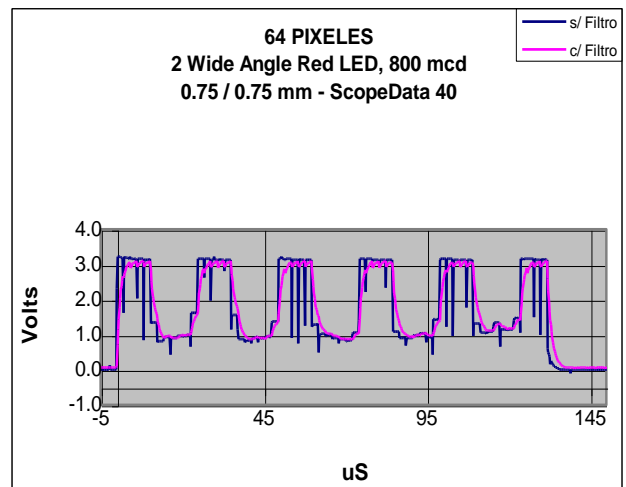
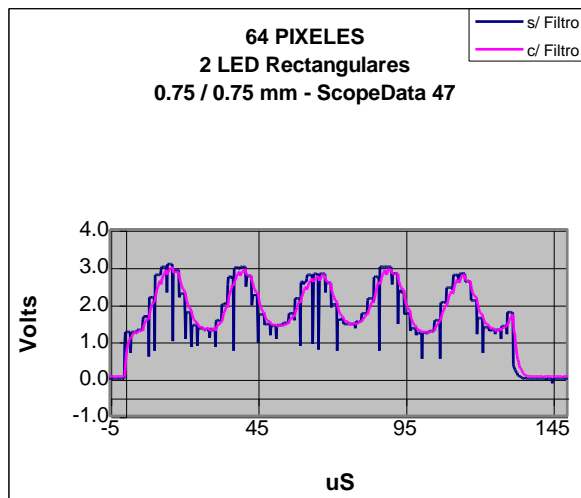
	Ensamblar el archivo actual
	Abrir programa de simulacion
	Abrir programa para programar
	Debugger en circuito (tarjeta del programador)
	Simulacion en circuito (tarjeta del programador)
	Cortar
	Copiar
	Pegar
	Abrir archivo (.ASM)
	Salvar archivo (.ASM)
	Salvar proyecto
	Cierra el archivo
	Permite visualizar los registros de cada modulo

PAINT

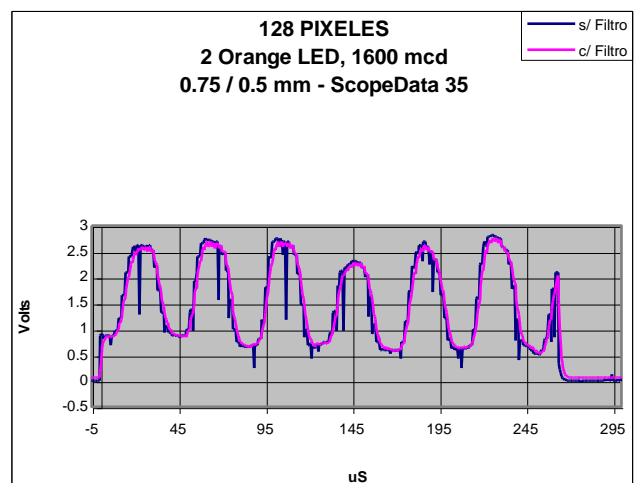
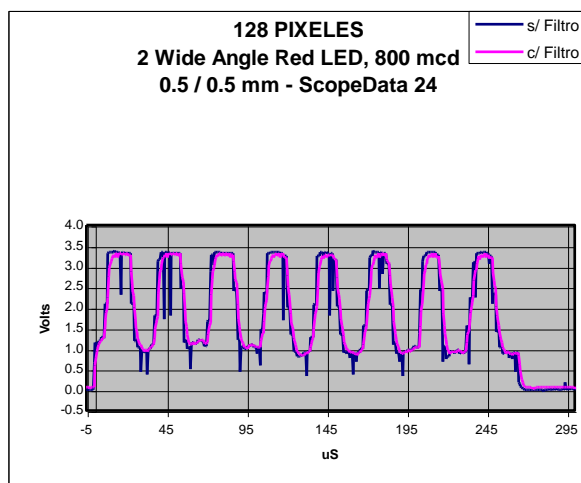
Apéndice 3: Gráficos de diferentes sensores con diferentes leds

Comparación de mediciones utilizando el mismo sensor de 64 píxeles, diferentes leds y diferentes anchos de barras.

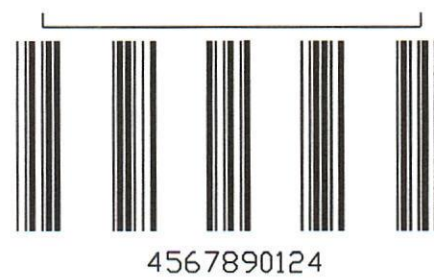
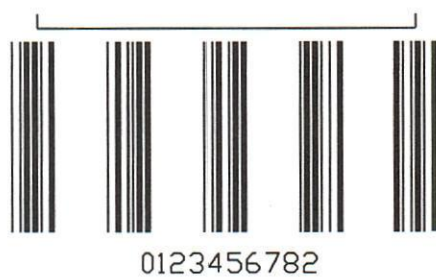
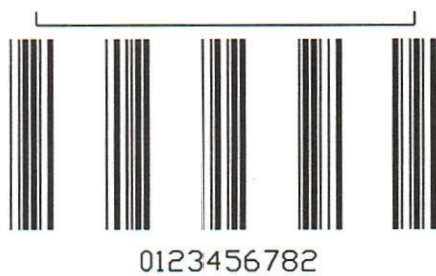
Estas gráficas se obtuvieron del osciloscopio HP 54645D, que trae un interfaz con Excel.



Comparación de mediciones utilizando el mismo sensor de 128 píxeles, diferentes leds y diferentes anchos de barras.



Apéndice 5: Muestras reales de tarjetas



AUTOCAD

Apéndice 5: Glosario

CCD: Charge Couple Device. Tipo de sensor especial para lecturas de código de barras por medio de píxeles.

Código de barras: Código basado en la representación de barras claras y oscuras, gruesas y delgadas.

Emulador: Dispositivo que simula en tiempo real el funcionamiento del microcontrolador ejecutando el programa seleccionado.

Enmascarable: Tipo de interrupción que puede ser o no atendida por medio del software.

ITF: Formato de código de barras que permite almacenar la mayor cantidad de información en el menor espacio posible.

LCD: Liquid Cristal Display. Display de cristal líquido. Dispositivo utilizado para representar caracteres en una pantalla.

Maquila: Fabrica que se dedica a ensamblar.

PLL: Phase lock loop. Lazo de fase cerrada. Circuito de control utilizado para estabilizar una frecuencia.

Transceiver: Dispositivo encargado de la transmisión y recepción de datos.

UART: Unidad asincrónica de recepción y transmisión de datos.

ANEXOS

Anexo 1: Características del transceiver LTC1487

Ultra-Low Power RS485 with Low EMI, Shutdown and High Input Impedance

FEATURES

- **High Input Impedance: Up to 256 Transceivers on the Bus**
- **Low Power: $I_{CC} = 120\mu A$ Max with Driver Disabled**
- $I_{CC} = 200\mu A$ Max with Driver Enabled, No Load
- **$1\mu A$ Quiescent Current in Shutdown Mode**
- Controlled Slew Rate Driver for Reduced EMI
- Single 5V Supply
- ESD Protection to $\pm 10kV$ On Receiver Inputs and Driver outputs
- $-7V$ to $12V$ Common-Mode Range Permits $\pm 7V$ Ground Difference Between Devices on the Data Line
- Thermal Shutdown Protection
- Power Up/Down Glitch-Free Driver Outputs Permit Live Insertion or Removal of Transceiver
- Driver Maintains High Impedance in Three-State or with the Power Off
- Pin Compatible with the LTC485

APPLICATIONS

- Battery-Powered RS485/RS422 Applications
- Low Power RS485/RS422 Transceiver
- Level Translator

DESCRIPTION

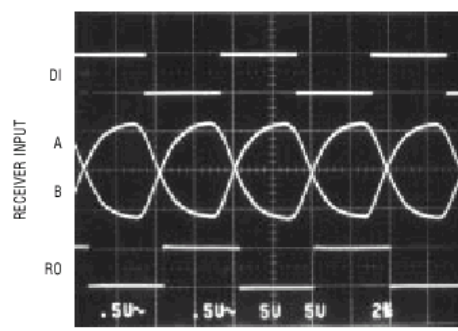
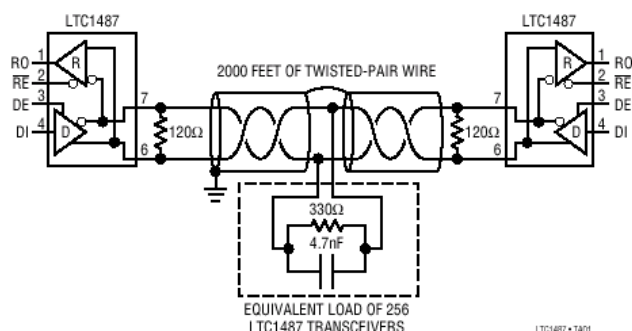
The LTC[®]1487 is an ultra-low power differential line transceiver designed with high impedance inputs allowing up to 256 transceivers to share a single bus. It meets the requirements of RS485 and RS422. The LTC1487 features output drivers with controlled slew rate, decreasing the EMI radiated from the RS485 lines, and improving signal fidelity with miterminated lines. The CMOS design offers significant power savings without sacrificing ruggedness against overload or ESD damage. Typical quiescent current is only $80\mu A$ while operating and $1\mu A$ in shutdown.

The driver and receiver feature three-state outputs, with the driver outputs maintaining high impedance over the entire common-mode range. Excessive power dissipation caused by bus contention or faults is prevented by a thermal shutdown circuit which forces the driver outputs into a high impedance state. The receiver has a fail-safe feature which guarantees a high output state when the inputs are left open. I/O pins are protected against multiple ESD strikes of over $\pm 10kV$ using the Human Body Model.

The LTC1487 is fully specified over the commercial temperature range and is available in 8-pin DIP and SO packages.

 LTC and LT are registered trademarks of Linear Technology Corporation.

TYPICAL APPLICATION



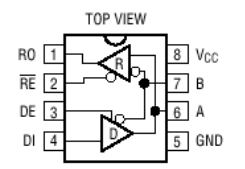
LTC1487

ABSOLUTE MAXIMUM RATINGS

(Note 1)

Supply Voltage (V_{CC})	12V
Control Input Voltage	$-0.5V$ to $V_{CC} + 0.5V$
Driver Input Voltage	$-0.5V$ to $V_{CC} + 0.5V$
Driver Output Voltage	$\pm 14V$
Receiver Input Voltage	$\pm 14V$
Receiver Output Voltage	$-0.5V$ to $V_{CC} + 0.5V$
Operating Temperature Range	$0^{\circ}C \leq T_A \leq 70^{\circ}C$
Lead Temperature (Soldering, 10 sec)	$300^{\circ}C$

PACKAGE/ORDER INFORMATION

 <p>N8 PACKAGE 8-LEAD PDIP</p> <p>S8 PACKAGE 8-LEAD PLASTIC SO</p> <p>$T_{JMAX} = 125^{\circ}C$, $\theta_{JA} = 130^{\circ}C/W$ (N8) $T_{JMAX} = 125^{\circ}C$, $\theta_{JA} = 150^{\circ}C/W$ (S8)</p>	ORDER PART NUMBER
	LTC1487CN8 LTC1487CS8
	S8 PART MARKING
	1487

Consult factory for Industrial and Military grade parts.

ELECTRICAL CHARACTERISTICS


$0^{\circ}C \leq T_A \leq 70^{\circ}C$, $V_{CC} = 5V$ (Notes 2, 3) unless otherwise noted.

SYMBOL	PARAMETER	CONDITIONS	MIN	TYP	MAX	UNITS
V_{OD1}	Differential Driver Output Voltage (Unloaded)	$I_O = 0$	●		5	V
V_{OD2}	Differential Driver Output Voltage (with Load)	$R = 50\Omega$ (RS422) $R = 27\Omega$ (RS485), Figure 1	● ●	2.0 1.5	5	V
ΔV_{OD}	Change in Magnitude of Driver Differential Output Voltage for Complementary Output States	$R = 27\Omega$ or $R = 50\Omega$, Figure 1	●		0.2	V
V_{OC}	Driver Common-Mode Output Voltage	$R = 27\Omega$ or $R = 50\Omega$, Figure 1	●		3	V
$\Delta V_{OC} $	Change in Magnitude of Driver Common-Mode Output Voltage for Complementary Output States	$R = 27\Omega$ or $R = 50\Omega$, Figure 1	●		0.2	V
V_{IH}	Input High Voltage	DE, DI, \overline{RE}	●	2		V
V_{IL}	Input Low Voltage	DE, DI, \overline{RE}	●		0.8	V
I_{IN1}	Input Current	DE, DI, \overline{RE}	●		± 2	μA
I_{IN2}	Input Current (A, B)	DE = 0, $V_{CC} = 0V$ or $5.25V$, $V_{IN} = 12V$ DE = 0, $V_{CC} = 0V$ or $5.25V$, $V_{IN} = -7V$	● ●		0.30 -0.15	mA
V_{TH}	Differential Input Threshold Voltage for Receiver	$-7V \leq V_{CM} \leq 12V$	●	-0.2	0.2	V
ΔV_{TH}	Receiver Input Hysteresis	$V_{CM} = 0V$	●	45		mV
V_{OH}	Receiver Output High Voltage	$I_O = -4mA$, $V_{ID} = 200mV$	●	3.5		V
V_{OL}	Receiver Output Low Voltage	$I_O = 4mA$, $V_{ID} = -200mV$	●		0.4	V
I_{OZR}	Three-State (High Impedance) Output Current at Receiver	$V_{CC} = \text{Max}$, $0.4V \leq V_O \leq 2.4V$	●		± 1	μA
R_{IN}	Receiver Input Resistance	$-7V \leq V_{CM} \leq 12V$	●	70	96	k Ω
I_{CC}	Supply Current	No Load, Output Enabled No Load, Output Disabled	● ●	120 80	200 120	μA
I_{SHDN}	Supply Current in Shutdown Mode	DE = 0V, $\overline{RE} = V_{CC}$		1	10	μA
I_{OSD1}	Driver Short-Circuit Current, $V_{OUT} = \text{HIGH}$	$-7V \leq V_O \leq 12V$	●	35	250	mA
I_{OSD2}	Driver Short-Circuit Current, $V_{OUT} = \text{LOW}$	$-7V \leq V_O \leq 12V$	●	35	250	mA
I_{OSR}	Receiver Short-Circuit Current	$0V \leq V_O \leq V_{CC}$	●	7	85	mA

ELECTRICAL CHARACTERISTICS $-40^{\circ}\text{C} \leq T_A \leq 85^{\circ}\text{C}$, $V_{CC} = 5\text{V}$ (Note 4) unless otherwise noted.

SYMBOL	PARAMETER	CONDITIONS		MIN	TYP	MAX	UNITS
V _{OD1}	Differential Driver Output Voltage (Unloaded)	I _O = 0	●			5	V
V _{OD2}	Differential Driver Output Voltage (with Load)	R = 50Ω (RS422) R = 27Ω (RS485), Figure 1	● ●	2.0 1.5		5	V V
V _{OC}	Driver Common-Mode Output Voltage	R = 27Ω or R = 50Ω, Figure 1	●			3	V
V _{TH}	Differential Input Threshold Voltage for Receiver	−7V ≤ V _{CM} ≤ 12V	●	−0.2		0.2	V
ΔV _{TH}	Receiver Input Hysteresis	V _{CM} = 0V	●		45		mV
I _{CC}	Supply Current	No Load, Output Enabled No Load, Output Disabled	● ●		120 80	200 120	μA μA
I _{SHDN}	Supply Current in Shutdown Mode	DE = 0V, $\overline{\text{RE}} = V_{\text{CC}}$			1	10	μA
t _{PLH}	Driver Input to Output	R _{DIFF} = 54Ω, C _{L1} = C _{L2} = 100pF, (Figures 3, 5)	●	150		1200	ns
t _{PHL}	Driver Input to Output		●	150		1200	ns
t _{SKEW}	Driver Output to Output		●		100	600	ns
t _r , t _f	Driver Rise or Fall Time		●	150		2000	ns
t _{PLH}	Receiver Input to Output	R _{DIFF} = 54Ω, C _{L1} = C _{L2} = 100pF, (Figures 3, 7)	●	30	140	250	ns
t _{PHL}	Receiver Input to Output		●	30	140	250	ns
t _{SKD}	t _{PLH} − t _{PHL} Differential Receiver Skew		●		13		ns
f _{MAX}	Maximum Data Rate		●	250			kbps

SWITCHING CHARACTERISTICS $0^{\circ}\text{C} \leq T_A \leq 70^{\circ}\text{C}$, $V_{CC} = 5\text{V}$ (Notes 2, 3) unless otherwise noted.

SYMBOL	PARAMETER	CONDITIONS		MIN	TYP	MAX	UNITS
t _{PLH}	Driver Input to Output	R _{DIFF} = 54Ω, C _{L1} = C _{L2} = 100pF, (Figures 3, 5)	●	150		1200	ns
t _{PHL}	Driver Input to Output		●	150		1200	ns
t _{SKEW}	Driver Output to Output		●		250	600	ns
t _r , t _f	Driver Rise or Fall Time		●	150		1200	ns
t _{ZH}	Driver Enable to Output High	C _L = 100pF (Figures 4, 6), S2 Closed	●	100		1500	ns
t _{ZL}	Driver Enable to Output Low	C _L = 100pF (Figures 4, 6), S1 Closed	●	100		1500	ns
t _{LZ}	Driver Disable Time from Low	C _L = 15pF (Figures 4, 6), S1 Closed	●	150		1500	ns
t _{HZ}	Driver Disable Time from High	C _L = 15pF (Figures 4, 6), S2 Closed	●	150		1500	ns
t _{PLH}	Receiver Input to Output	R _{DIFF} = 54Ω, C _{L1} = C _{L2} = 100pF, (Figures 3, 7)	●	30	140	250	ns
t _{PHL}	Receiver Input to Output		●	30	140	250	ns
t _{SKD}	t _{PLH} – t _{PHL} Differential Receiver Skew		●		13		ns
t _{ZL}	Receiver Enable to Output Low	C _{RL} = 15pF (Figures 2, 8), S1 Closed	●		20	50	ns
t _{ZH}	Receiver Enable to Output High	C _{RL} = 15pF (Figures 2, 8), S2 Closed	●		20	50	ns
t _{LZ}	Receiver Disable from Low	C _{RL} = 15pF (Figures 2, 8), S1 Closed	●		20	50	ns
t _{HZ}	Receiver Disable from High	C _{RL} = 15pF (Figures 2, 8), S2 Closed	●		20	50	ns
f _{MAX}	Maximum Data Rate		●	250			kbps
t _{SHDN}	Time to Shutdown	DE = 0, RE = 	●	50	200	600	ns

LTC1487

SWITCHING CHARACTERISTICS $0^{\circ}\text{C} \leq T_A \leq 70^{\circ}\text{C}$, $V_{CC} = 5\text{V}$ (Notes 2, 3) unless otherwise noted.

SYMBOL	PARAMETER	CONDITIONS	MIN	TYP	MAX	UNITS
$t_{ZH}(\text{SHDN})$	Driver Enable from Shutdown to Output High	$C_L = 100\text{pF}$ (Figures 4, 6), S2 Closed	●		2000	ns
$t_{ZL}(\text{SHDN})$	Driver Enable from Shutdown to Output Low	$C_L = 100\text{pF}$ (Figures 4, 6), S1 Closed	●		2000	ns
$t_{ZH}(\text{SHDN})$	Receiver Enable from Shutdown to Output High	$C_L = 15\text{pF}$ (Figures 2, 8), S2 Closed	●		2000	ns
$t_{ZL}(\text{SHDN})$	Receiver Enable from Shutdown to Output Low	$C_L = 15\text{pF}$ (Figures 2, 8), S1 Closed	●		2000	ns

The ● denotes specifications which apply over the full operating temperature range.

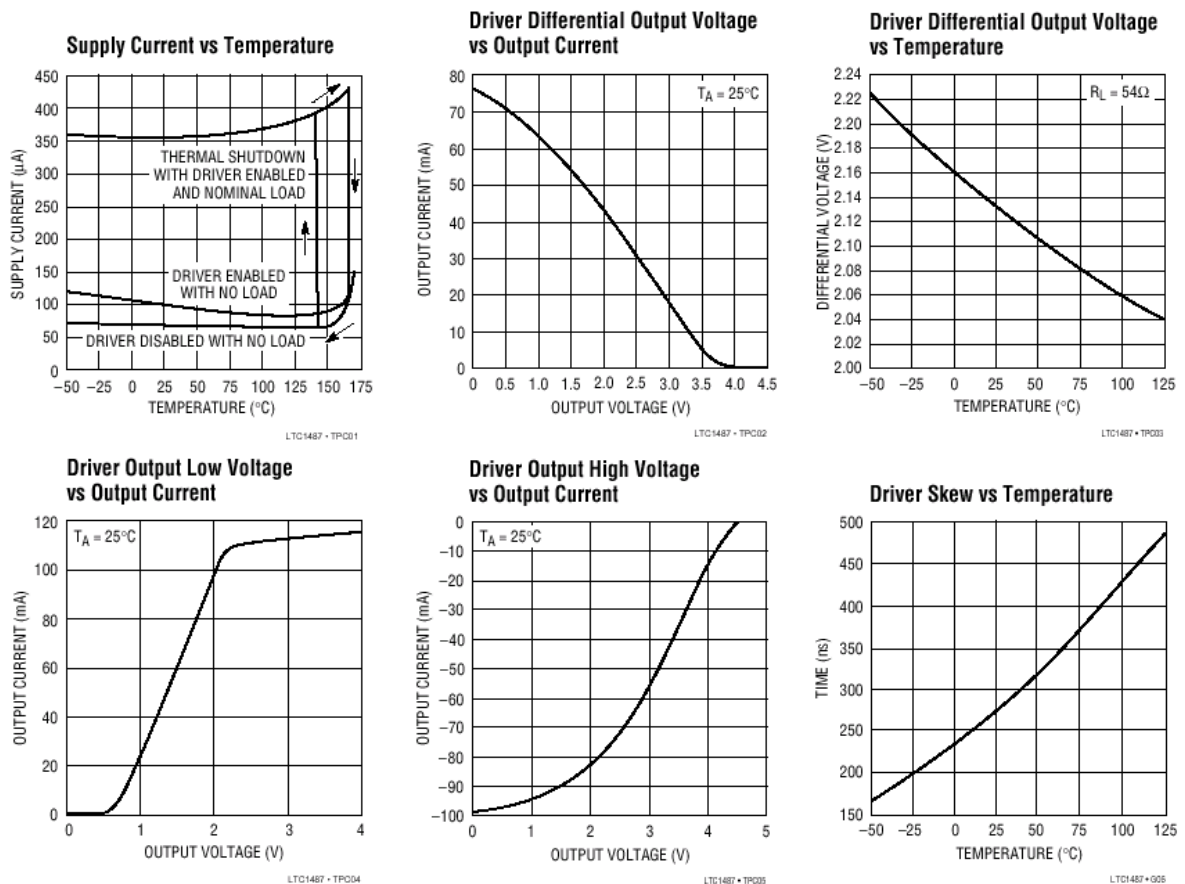
Note 1: Absolute maximum ratings are those beyond which the safety of the device cannot be guaranteed.

Note 2: All currents into device pins are positive; all currents out of device pins are negative. All voltages are referenced to device ground unless otherwise specified.

Note 3: All typicals are given for $V_{CC} = 5\text{V}$ and $T_A = 25^{\circ}\text{C}$.

Note 4: The LTC1487 is not tested and is not quality-assurance sampled at -40°C and at 85°C . These specifications are guaranteed by design, correlation, and/or inference from 0°C , 25°C and/or 70°C tests.

TYPICAL PERFORMANCE CHARACTERISTICS



PIN FUNCTIONS

RO (Pin 1): Receiver Output. If the receiver output is enabled (\overline{RE} LOW), and $A > B$ by 200mV, RO will be HIGH. If $A < B$ by 200mV, then RO will be LOW.

\overline{RE} (Pin 2): Receiver Output Enable. A LOW enables the receiver output, RO. A HIGH input forces the receiver output into a high impedance state.

DE (Pin 3): Driver Outputs Enable. A HIGH on DE enables the driver output. A and B and the chip will function as a line driver. A LOW input will force the driver outputs into a high impedance state and the chip will function as a line receiver. If \overline{RE} is HIGH and DE is LOW, the part will enter a low power (1 μ A) shutdown state.

DI (Pin 4): Driver Input. If the driver outputs are enabled (DE HIGH) then a LOW on DI forces the outputs A LOW and B HIGH. A HIGH on DI with the driver outputs enabled will force A HIGH and B LOW.

GND (Pin 5): Ground.

A (Pin 6): Driver Output/Receiver Input.

B (Pin 7): Driver Output/Receiver Input.

V_{CC} (Pin 8): Positive Supply. $4.75V < V_{CC} < 5.25V$.

FUNCTION TABLES

LTC1487 Transmitting

INPUTS			OUTPUTS	
\overline{RE}	DE	DI	B	A
X	1	1	0	1
X	1	0	1	0
0	0	X	Z	Z
1	0	X	Z*	Z*

*Shutdown mode

LTC1487 Receiving

INPUTS			OUTPUTS
\overline{RE}	DE	A – B	RO
0	0	$\geq 0.2V$	1
0	0	$\leq -0.2V$	0
0	0	Inputs Open	1
1	0	X	Z*

*Shutdown mode

TEST CIRCUITS

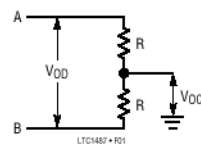


Figure 1. Driver DC Test Load

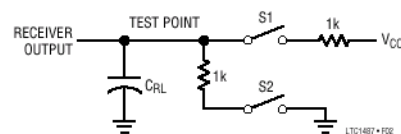


Figure 2. Receiver Timing Test Load

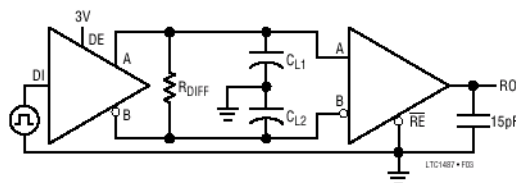


Figure 3. Driver/Receiver Timing Test Circuit

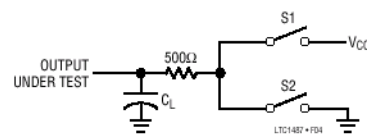


Figure 4. Driver Timing Test Load

SWITCHING TIME WAVEFORMS

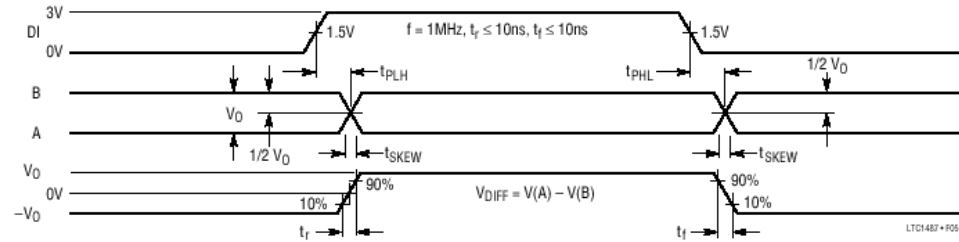


Figure 5. Driver Propagation Delays

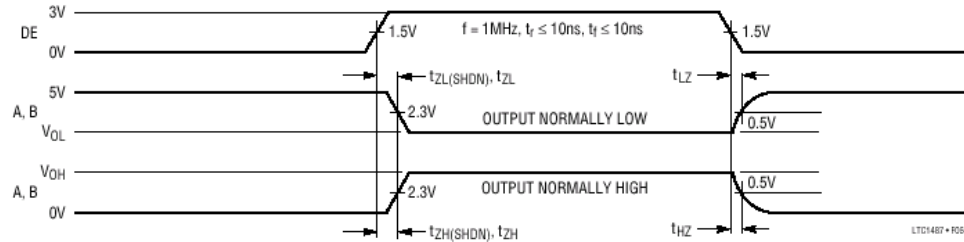


Figure 6. Driver Enable and Disable Times

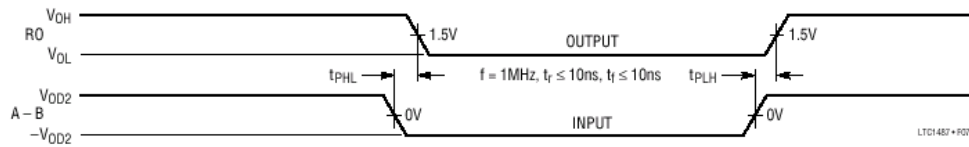


Figure 7. Receiver Propagation Delays

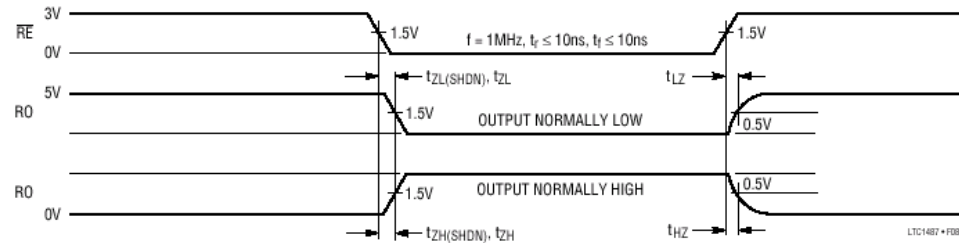


Figure 8. Receiver Enable and Disable Times

APPLICATIONS INFORMATION

High Input Impedance

The LTC1487 is designed with a $96\text{k}\Omega$ (typ) input impedance to allow up to 256 transceivers to share a single RS485 differential data bus. The RS485 specification requires that a transceiver be able to drive as many as 32 "unit loads." One unit load (UL) is defined as an impedance that draws a maximum of 1mA with up to 12V across it. Typical RS485 transceivers present between 0.5 and 1 unit load at their inputs. The $96\text{k}\Omega$ input impedance of the LTC1487 will draw only 125 μA under the same 12V condition, presenting only 0.125UL to the bus. As a result, 256 LTC1487 transceivers ($32\text{UL}/0.125\text{UL} = 256$) can be connected to a single RS485 data bus without exceeding the RS485 driver load specification. The LTC1487 meets all other RS485 specifications, allowing it to operate equally well with standard RS485 transceiver devices or high impedance transceivers.

CMOS Output Driver

The RS485 specification requires that a transceiver withstand common-mode voltages of up to 12V or -7V at the RS485 line connections. Additionally, the transceiver must be immune to both ESD and latch-up. This rules out traditional CMOS drivers, which include parasitic diodes from their driver outputs to each supply rail (Figure 9). The LTC1487 uses a proprietary process enhancement which adds a pair of Schottky diodes to the output stage (Figure 10), preventing current from flowing when the common-mode voltage exceeds the supply rails. Latch-up at the output drivers is virtually eliminated and the driver is prevented from loading the line under RS485 specified fault conditions. A proprietary output protection structure protects the transceiver line terminals against ESD strikes (Human Body Model) of up to $\pm 10\text{kV}$.

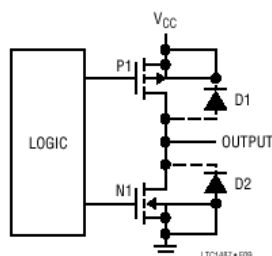


Figure 9. Conventional CMOS Output Stage

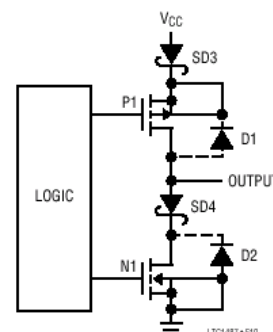


Figure 10. LTC1487 Output Stage

When two or more drivers are connected to the same transmission line, a potential condition exists whereby more than two drivers are simultaneously active. If one or more drivers is sourcing current while another driver is sinking current, excessive power dissipation may occur within either the sourcing or sinking element. This condition is defined as driver contention, since multiple drivers are competing for one transmission line. The LTC1487 provides a current limiting scheme to prevent driver contention failure. When driver contention occurs, the current drawn is limited to about 70mA, preventing excessive power dissipation within the drivers.

The LTC1487 has a thermal shutdown feature which protects the part from excessive power dissipation. Under extreme fault conditions, up to 250mA can flow through the part, causing rapid internal temperature rise. The thermal shutdown circuit will disable the driver outputs when the internal temperature reaches 150°C and turns them back on when the temperature cools to 130°C . This cycle will repeat as necessary until the fault condition is removed.

Receiver Inputs

The LTC1487 receiver features an input common-mode range covering the entire RS485 specified range of -7V to 12V . Internal $96\text{k}\Omega$ input resistors from each line terminal to ground provide the 0.125UL load to the RS485 bus. Differential signals of greater than $\pm 200\text{mV}$ within the specified input common-mode range will be converted to a TTL-compatible signal at the receiver output. A small amount of input hysteresis is included to minimize the

LTC1487

APPLICATIONS INFORMATION

effects of noise on the line signals. If the line is terminated or the receiver inputs are shorted together, the receiver output will retain the last valid line signal due to the 45mV of hysteresis incorporated in the receiver circuit. If the LTC1487 transceiver inputs are left floating (unterminated), an internal pull-up of 10 μ A at the A input will force the receiver output to a known high state.

Low Power Operation

The LTC1487 draws very little supply current whenever the driver outputs are disabled. In shutdown mode, the quiescent current is typically less than 1 μ A. With the receiver active and the driver outputs disabled, the LTC1487 will typically draw 80 μ A quiescent current. With the driver outputs enabled but unterminated, quiescent current will rise slightly as one of the two outputs sources current into the internal receiver input resistance. With the minimum receiver input resistance of 70k and the maximum output swing of 5V, the quiescent current will rise by a maximum of 72 μ A. Typical quiescent current rise with the driver enabled is about 40 μ A.

The quiescent current rises significantly if the driver is enabled when it is externally terminated. With 1/2 termination load (120 Ω between the driver outputs), the quiescent current will jump to at least 13mA as the drivers force a minimum of 1.5V across the termination resistance. With a fully terminated 60 Ω line attached, the current will rise to greater than 25mA with the driver enabled, completely overshadowing the extra 40 μ A drawn by the internal receiver inputs.

Shutdown Mode

Both the receiver output (RO) and the driver outputs (A, B) can be placed in three-state mode by bringing \overline{RE} HIGH and DE LOW respectively. In addition, the LTC1487 will enter shutdown mode when \overline{RE} is HIGH and DE is LOW.

In shutdown the LTC1487 typically draws only 1 μ A of supply current. In order to guarantee that the part goes into shutdown, \overline{RE} must be HIGH and DE must be LOW for at least 600ns simultaneously. If this time duration is less than 50ns the part will not enter shutdown mode. Toggling either \overline{RE} or DE will wake the LTC1487 back up within 3.5 μ s.

If the driver is active immediately prior to shutdown, the supply current will not drop to 1 μ A until the driver outputs have reached a steady state; this can take as long as 2.6 μ s under worst case conditions. If the driver is disabled prior to shutdown the supply current will drop to 1 μ A immediately.

Slew Rate and Propagation Delay

Many digital encoding schemes are dependent upon the difference in the propagation delay times of the driver and receiver. Figure 11 shows the test circuit for the LTC1487 propagation delay.

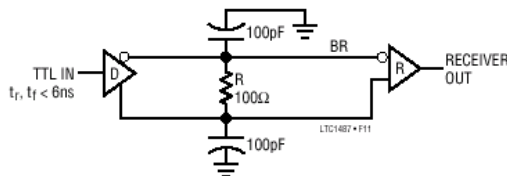


Figure 11. Receiver Propagation Delay Test Circuit

The receiver delay times are:

$$|t_{PLH} - t_{PHL}| = 13\text{ns Typ, } V_{CC} = 5\text{V}$$

The LTC1487 drivers feature controlled slew rate to reduce system EMI and improve signal fidelity by reducing reflections due to misterminated cables.

The driver's skew times are:

$$\text{Skew} = 250\text{ns Typ, } V_{CC} = 5\text{V}$$

$$600\text{ns Max, } V_{CC} = 5\text{V, } T_A = -40^\circ\text{C to } 85^\circ\text{C}$$

PACKAGE DESCRIPTION

For package descriptions consult the 1994 Linear Databook Volume III.